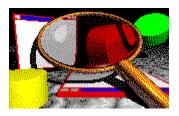# Project Analyzer 3.0

Overview of Project Analyzer

Introduction
How it works

Version history - What's new?

Registration, future versions and contact information

Menus

File

Open project...
-
Generate constants module...
Archive project files...
-
Exit

View

Find procedure...
-
File details
Procedure details
Variable and constant list
-
Cyclomatic complexities...
Nested loops...
-
File call tree
Procedure call tree
Hypertext

Report

File details
Procedure details
-
File call tree
Procedure call tree
    All procedures
    Selected file only
    Selected procedure only
-
List files...
List procedures...

List variables and constants...
Needless globals report
DLL report
-
Problem report
Project summary
Options
  Show report on double click
  Show detail window on double click
  -
  Report to
    Display
    Printer
    File...
    Editor...
  Printer setup...
  Printer font...
  -
  Enhanced display output
  Dull gray windows
  RTF Editor is MS Word
  -
  Hypertext options...

Project Analyzer was written by **Tuomas.Salste@Helsinki.Fi.**

Visit the Visual Basic Shop WWW page
http://www.helsinki.fi/~salste/vb.html
Shareware Visual Basic tools

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

List files

Product Analyzer can produce list of files that belong to your project. To get a list, choose **Report|List files**. A dialog box will appear.

You can get lists sorted by:
• File name
• File type

If you want just a list of file names, leave the **Details** box unchecked. If you check it, the list will contain the following additional data:
• Comments
• Which files need procedures in this file
• Which files this file needs

Note: You get the commented lines in the (declarations) section of your .BAS, .FRM, .TXT or .GLB file. The program shows those comment lines that come before the first non-comment line, like this:

```
' This is a function module
' These comments will be reported

Dim Text As String
' This comment will not be reported
```

The file list goes to the <u>output</u> device defined with the **Output goes to** radio buttons.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

List procedures

Product Analyzer can produce list of procedures that belong to your project. To get a list, click the **List procedures** button or choose **Report|List procedures**. A dialog box will appear.

You can select if you want the following data included in the procedure list or not:
- Comments
- Which procedures use this procedure
- Which procedures this procedure uses

In the registered version there are also options to list procedures that are or are not needed by your program. This is useful for finding **dead code**.

Note on comments: The program will show those commented lines at the beginning of each sub/function that come before the first empty or non-comment line. Commented lines before and after the sub/function declaration line are shown, as well as procedure description (VB 4.0), if available. An example:

```
' This sub was created by N.N.
SUB MySub (Byval x As Integer)
Attribute VB_Description=Prints x + 5 on the form
' This sub does the following:
' It takes the parameter x and ...

' This line is not shown any more
Form1.Print x + 5

END SUB
```

The procedure list goes to the output device defined with the **Output goes to** radio buttons.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Archive Project Files

Project Analyzer helps backing up your program files. Currently it supports two archiver programs, namely **ARJ** (current version: 2.41a) and **PKZIP** (version at this time: 2.04g). You can also easily build file lists of your project files in order to manually copy/archive them without effort.

Pack Project Files Dialog

Select files to be packed (or files whose file name are to be saved in the list file) in the list box. By default, all VB files belonging to your project are selected, but no DLLs. These are the files that you most probably want to save.

If you want to add files to the list you can click the Add button. Files ending in **.lst** are considered to be list files that contain a list of more files you'd want to add to the list box. You can create these .lst files with any text editor or with Project Analyzer itself.

The **ARJ Command** and **PKZIP Command** fields contain the commands and command switches used to create new archive files.

Sample ARJ Command: ARJ a MYARCH.ARJ !MYPROJ.LST
Sample PKZIP Command: PKZIP MYARCH.ZIP @MYPROJ.LST

You only have to supply the blue part of the command line; Project Analyzer does the rest. Now you can pack by pressing one of the **Pack** buttons.

By default, a temporary file is used to tell the archiver program what files to pack. If you want to save that list file for later use you can click the **Select** button.

Having named a list file, you can save the list file only if you want. In other words, a new button name **List files only** will appear.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Introduction

Project Analyzer is a Finnish <u>shareware</u> tool designed to help creating, maintaining and reporting applications developed in Visual Basic 3.0 and 4.0. The program should work with VB 2.0 too.

Project Analyzer makes a full, two-phase source code analysis. It collects detailed information about all the files, procedures, controls, variables, constants, classes etc. in the VB project. The results can be viewed on the screen, in textual format in your favourite word processor, or on paper.

What is needed to run Project Analyzer

Project Analyzer 3.0 needs Visual Basic 4.0 DLLs to run. You can run it with Visual Basic 3.0, but you will then have the DLLs instead. Visit my WWW site (**http://www.helsinki.fi/~salste/vb.html**) to download the needed DLLs.

Project Analyzer is available in both 16-bit and 32-bit versions. **You only need one version** for analyzing VB 3.0, VB 4.0 (16-bit) and VB 4.0 (32-bit) projects. If you are only using VB 3.0 get the 16-bit version of Project Analyzer, I only distribute the 16-bit DLLs.

Features

Full source code analysis
- <u>File details</u>: procedure and control lists, file level cross-references
- <u>Procedure details</u>: cyclomatic complexity, lines of code, procedure level cross-references
- Find dead or live procedures, variables and constants. Remove dead constants.
- Variable and constant declaration information, both dead and live
- <u>Call trees</u>
- <u>Cyclomatic complexity</u>

Reports for maintaining the project
- <u>Sub and Function lists</u> with comments for reporting purposes
- <u>File lists</u> with comments
- <u>DLL reports</u>
- <u>Problem report</u>
- <u>Project summary report</u>
- <u>Variable and constant lists</u> with dead or live variables & constants

Other powerful features
- Project file <u>archiving</u> with PKZIP or ARJ
- <u>Hypertext</u> style code viewer
- FRX file view

See <u>version history</u> for new features.

Why? Why???

Maintain phase costs rule the software industry. When developing applications, especially those that someone else has to maintain, it's important to keep track on what each sub/function does, and what other procedures they need to work.

This is where Project Analyzer can help you. With it you can produce detailed cross-reference reports with supplied comments. You can view call trees. And you can find dead code, dead

variables, and dead constants.

Maintaining programs is **hell**. But Project Analyzer will turn it into **heaven** :-)

How to use it?

Using Project Analyzer is simple. Open a **.mak** or **.vbp** file with the Open Project command in the File menu and wait. Project Analyzer will collect information about the **.bas**, **.frm**, **.frx, .cls,** and **.dll** files in the background.

Note: If you are using VB 3.0, remember to save all files in your project in text format.

The analysis consist of two phases. The first phase collects the most data, and the second one will check for cross-references and find dead code etc. If a menu command or a button is greyed, don't panic, just wait for the analysis to end.

Hint: Test Project Analyzer by opening the included TEST30.MAK or TEST40.VBP file (named after VB30 and VB40). These are pure demo projects for you to see how Project Analyzer works. You can of course use your own projects too for this purpose.

When the analysis has ended, just double click a procedure or a file to see the results. Remember to push the **Hypertext** button. Don't forget to examine the View and Report menus either, many of the most powerful features reside there. Feel free to test your own projects too - Project Analyzer will not modify them!

Registration
Collecting information
Back to Contents

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Collecting Project Information

When you start Project Analyzer, or when you choose Open Project... from the File menu, you get a dialog box asking you to choose a project file (a VB **.mak** or **.vbp** file).

Project Analyzer examines the project file to find out what files belong to your project. After determining the type of each file, the program further looks inside these files.

What information Project Analyzer searches for in the files:

Project Analyzer examines the **.bas, .frm, .frx, .cls, .dll** and other files belonging to your project. As an exception, Project Analyzer doesn't analyze .vbx, .ocx nor .res files.

If you use VB 3.0, remember to save the files in text format. Project Analyzer can't read VB files saved in binary format, other than the .frx files. It will analyze .dll files based on the declarations in your basic code. - If the program finds a binary file it will simply ignore its contents. This will result in incomplete analysis, like showing dead variables that aren't dead in reality.

Project Analyzer builds a list of all files and procedures (i.e. subs and functions) that your project consists of. Code is analyzed at form and procedure level to list the controls and the variables & constants. Furthermore, procedure and variable & constant cross-references are examined.

Note: Information is collected in the background. So, when Project Analyzer is examining your files you can do other things like running other programs - or you can view the information the program has collected so far. There are some things you can't do when the collecting is in progress, like getting a list of all procedure calls - that's because the program hasn't yet finished collecting that piece of information!

The program reads your files twice. You'll notify when it's ready by looking for the word "Ready!" at the lower left corner of the program window. Should you want to stop the process, you can press the Stop button. To speed up the collecting process, the program checks if you have pressed the Stop button only every now and then, so be patient!

Use Option Explicit. You will get better results if you're declaring all your variables. This is good practice anyway.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Report Formats

Project Analyzer can currently produce reports in various different ways. To select one of them, select the **Options|Report to** menu command or one of the **Report to** radio buttons.

Depending on what you select, you either get a formatted report, or a text-only version.

1. To the display
This is the default. The output goes to a special Display window where you can read the text. You have 2 options to take:
1. A formatted report
2. An unformatted, plain text report (from which you can copy parts to the clipboard)
The type of report you get is determined by the **Options|Enhanced display output** menu option.

There are 3 of these Display windows available. You can use them side-by-side.

2. To the printer
If you want, you can choose the output to go to the selected Windows default printer. This report is formatted.

To select a printer, click **Options|Printer setup**. To choose a printer font, click **Options| Printer font**.

3. To a file
Project Analyzer can currently direct its output to:
1) a Windows text file (plain text)
2) a MS Windows Write .wri file (plain text)
3) a Rich Text Format (RTF) file (formatted report)

4. To an editor
This is similar to the 3rd option, but after saving the report to a file, Project Analyzer launches an editor to view the report.   You may use Notepad, Write,   WordPad, MS Word, or any text editor you like.

To run your text editor from Project Analyzer you should associate the file name extension (**.txt**, **.wri** or **.rtf**) with your editor. (by default, .txt files are associated with Notepad and .wri files with Write or WordPad).

Note: When producing large outputs (over 32kB in size) you can get an "Out of string space" error when the output is going to the unformatted display window, or to a Write file. The report will be truncated to 32 kB. This usually occurs only when you're trying to get a procedure list with details (these lists can be horribly long!). The solution is to use the enhanced display, printer, text file or RTF file option, which allow reports of any length.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

About Project Analyzer

Contact me!

I'm very eager to get any comments about Project Analyzer. I'd like to get your comments even if you don't register and even if you didn't like Project Analyzer at all. My Internet address is **Tuomas.Salste@Helsinki.Fi.**

You may also want to visit my VBShop WWW page **http://www.helsinki.fi/~salste/vb.html** Product information will be available through that page. You will also find information about my other shareware and freeware products there.

New versions

Project Analyzer is still under further development. New versions with further enhancements can be expected in the future. New features will mainly be available in the registered version only.

I will notify all registered users about any new versions. Registered users will get the new versions for free. Distribution of the updates is done solely on the Internet, usually by WWW.

Registration

The following text applies to **unregistered** users only.

Project Analyzer is **shareware.** If you use this program for anything else than evaluation purposes, you'll need to pay a small registration fee. When you pay the fee, you get the full, registered version of Project Analyzer with more features. You will also get **all future versions for free**. *Getting new versions for free requires an Internet email address.*

Be sure the read the following:
1. The general registration instructions in REGISTRA.HLP. They apply to all my Visual Basic tools.
2. The special registration instructions below. They apply especially to Project Analyzer.

Price

The price for registering Project Analyzer 3.0 is:
USD   80   USA
FIM   350   Finland

You can also register in your own currency. You can pay in any currency not in the list, but please negotiate with me first. Examples of the prices are listed below, these are based on rates of 22 March 1996. The below prices are valid provided there are no drastical changes in the exchange rates.

GBP   55   Great Britain
DEM 120   Germany
SEK   550   Sweden
NOK 500   Norway

The source code is also available for an extra fee, as well as multiuser licenses. Email me and ask more.

Payment method

The possible payment methods are described in REGISTRA.HLP.

Registration keyword

After registering, you will get a registration keyword that will enable the registered features. Input this keyword when Project Analyzer starts, and you will be registered.

Other utilities for Visual Basic:

•DBtoVB Wizard - automatically generate functions for creatíng a new database - no coding!
•DB Lock - examine the behaviour and locking scheme of your multi-user database application
•DB Structure - print the structure of an MS Access database

See the Visual Basic shop WWW page (**http://www.helsinki.fi/~salste/vb.html**) or email to me for more information!

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Hypertext

(version 1.1 onwards, <u>registered version</u> only)

You can see your code in hypertext form (like text in this help file) by using Project Analyzer's hypertext feature. The hypertext window is a read-only code window but with additional features. You can see called procedures easily by clicking at the underlined procedure names. If a constant or variable is <u>underlined</u>, you can jump to the declarations section where it was declared.

To open the hypertext window, select a file or a procedure in the **Files** or **Procedures** list box and either press the **Hypertext** button or select **View|Hypertext** hypertext in the menus. Another method is to press the **Control** key down and then click in the **Files** or **Procedures** list box.

To see the procedures that call the shown procedure click the **Called by** button.

Try right-clicking highlighted variables, constants and procedures. You will get a menu with links to their definitions, and in the case of global and module-level definitions you will also see where they are referenced and assigned to!

Options
You can change the colors of different types of hotlinks by selecting **View|Options**. If you want, Project Analyzer can ~~**overstrike**~~ dead variables too.

Note: This feature is available in the registered version only. In the unregistered version you can only use it with small projects.

List variables and constants

Product Analyzer can produce list of the global and module-level variables and constants that belong to your project. To get a list, choose **Report|List variables and constants** in the main window, or press the **Report** button in the <u>Variables and constants window</u>. A dialog box will appear.

You can get two kinds of lists:
1. List all **global** variables and constants in alphabetical order
2. List all **global** and **module level** variables and constants ordered by file they appear in

The file list goes to the <u>output</u> device defined with the Output goes to radio buttons.

Note 1: Variables and constants information is available when analysis phase 1 is complete.

Note 2: You can get more versatile variable and constant lists from the <u>Variables and constants window</u>.

Version History

See a more detailed version history in VERSION.TXT.

Version 3.0 (Summer 1996):
- Added support for Visual Basic 4.0
- Project Analyzer now needs Visual Basic 4.0 DLLs to run. If you only use VB 3.0, please visit VBShop (**http://www.helsinki.fi/~salste/vb.html**) to download the required DLLs.
- Class modules supported
- Hypertext window enhanced
- Call tree enhanced
- Cyclomatic complexities view
- Nested loops view
- Needless globals report
- A lot of enhancements to various features

Version 2.1 (September 1995):
- Project summary report
- Generate constants module command
- Modifications to look better with Windows 95
- Enhancements in the hypertext window
- Average cyclomatic complexity (version 2.1.05 onwards, registered version only)
- About 25% faster analysis (appears in the second phase) due to improving PROJECT.DLL
- Formatted reports on the screen too (version 2.1.07 onwards)
- DLL Report (version 2.1.07 onwards, registered version only)
- Can handle larger projects than the previous versions

Version 2.0 (August 1995):
- The Display window is no more the only way of seeing the results of an analysis. Separate detail windows have been implemented.
- File details and Procedure details can now be viewed on the screen in a separate window in addition to the reports
- File details show the form's icon and controls too
- New Variables and constants window with information about references to them (dead variables & constants, referencing files)
- New Call tree window
- File level call trees added in addition to procedure level call trees
- FRX files can be examined too (just double click the FRX file in the main window)
- Shows declared functions in external DLLs
- Problem report (registered version only)
- Formatted reports using RTF files
- New statistics, like cyclomatic complexity (registered version only) and lines of code for each procedure
- Find command now in the Hypertext and Procedure details window too

Version 1.1 (March 1995):
- Find dead procedures (registered version only)
- Global and module level Variable and constant lists
- Procedural call trees (registered version only)
- Hypertext style code viewer (registered version only)
- Saving the results of the analysis for future use (registered version only)
- A **Find procedure** command to quickly locate a procedure

Version 1.0 (February 1995):
The initial release.

Call tree

(version 1.1 onwards, <u>registered version</u> only)

You can see the procedure and file dependencies in the form of a call tree. The tree is available in two formats:

1. Call tree in a separate window

Use the **View|Procedure call tree** and **View|File call tree** menu commands to see an expandable view of the dependencies. You can get a textual report of the tree by pressing the **Report** button.

2. Textual report

The reports are available using the **Report|Procedure call tree** and **Report|File call tree** menu commands. Three subtypes of a procedure call tree report are available:
2.1. All procedures
2.2. Procedures in selected file only
2.3. The selected procedure only

If a procedure calls itself or another procedure that then eventually call the first procedure back, it will be marked **<recursive call>** in the tree.

Note 1: This call tree features are available in the registered version only. In the unregistered version you can see the call tree only for small projects.

Note 2: The call trees are available after analysis phase 2 has completed.

Note 3: You can see procedural dependencies in <u>hypertext</u> or <u>list</u> form too.

Note 4: You may get an out of memory error when trying to see large call trees in the dedicated window. In this case, get the textual report instead.

If you don't like the background colors I've chosen for Project Analyzer, you can set them to gray using **Options|Dull gray windows**.

File details window

(version 2.0 onwards)

The **File details** window shows information about a specified file in your project. The easiest way to see File details is to double click the **Files** list box in the main window.

File details can show information about:
- Form files
- FRX files
- Basic modules
- Class modules
- DLL's

Basic files (FRM, BAS, CLS)

- Code lines
- Size in kilobytes
- Form icon (FRM files only)
- File version (FRM files only): The file format of the FRM file. Note that VB 3.0 still creates FRM version 2.0 files
- Subs and functions: Name, type (functions only), lines of code, cyclomatic complexity, and 'Dead' status if the procedure is not used by your project. Note that if there are question marks '?' in the cyclomatic complexity and 'Dead' columns it means that analysis phase 2 has not yet been completed. - Double click a procedure to see procedure details.
- Variables and constants defined in this file: Global and module-level definitions.
- Controls (FRM files only)
- List of files that call this file (BAS files only, after phase 2)
- List of files that this file calls (after phase 2)

Some of the information will not be available before the analysis phase 2 has been completed. To see the results after completion of analysis use the View|Refresh menu command.

DLL files (DLL, EXE, ...)

DLL's are analyzed like BAS files, except:
- Variables and constants are not listed
- Files that are called by this DLL are not analyzed
- The lines of code in a DLL cannot be analyzed
- DLL file version is not analyzed
Only those procedures that were declared with a Declare statement are shown.

FRX files

Project Analyzer can examine the contents of a FRX file that is there if you saved your FRM file in text format and the form contained pictures. The FRX file details window shows a list of the pictures that are in the FRX file, and the names of the controls these pictures are for.

Report and Hypertext

To see the procedure details as a textual listing, press the **Report** button. To see the selected procedure as hypertext, select the **View|Hypertext** menu command.

Note: VBX, OCX and RES files are analyzed by Project Analyzer.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Procedure details window

(version 2.0 onwards)

The **Procedure details** window shows information about a specified sub or function. The procedure can be a basic procedure, an event or a library procedure.

The following information is shown:
- Procedure name and type
- Procedure declaration
- Procedure file
- The 'dead' state of the procedure. This field is available only after analysis phase 2 has been completed, before that the text 'Not analyzed' is shown. To see the results after completion of analysis use the View|Refresh menu command.
- Lines of code (doesn't apply to library procedures)
- Cyclomatic complexity   (doesn't apply to library procedures, nor to declarations sections)
- Procedures that call this procedure
- Procedures that are called by this procedure (doesn't apply to library procedures)

To see the information as a textual listing, press the **Report** button. To see the entire procedure code, press the Hypertext button.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Variables and constants window

Project Analyzer can show a list of all global and module-level definitions either as a <u>textual report</u> or in a list box. To see the list box, either press the **Vars & consts** button in the main window, or select the **View|Variable and constant list** menu command.

You can choose from the following options:
- Show constants
- Show variables
- Show global definitions defined in the (declarations) section of a module
- Show module-level definitions defined in the (declarations) section of a form or module
- Sort by file
- Sort by name
- Show live definitions (those that are referenced somewhere in the program)
- Show dead definitions. Those ones are dead that are not referenced even once in the program, even though a dead variable may be assigned to.

To see the actual definition of a variable or a constant, press the **View declaration** button. That will bring up the <u>Hypertext window</u> with the corresponding (declarations) section. To view a <u>textual listing</u>, press the **Report** button.

To see which procedures reference or assign to the selected variable or constant, select the **View references** command. A list will show up with all assignments and references to that variable or constant.

Note 1: Variables and constants information is available when analysis phase 1 is complete. The reference statistics are available when phase 2 is complete. To display the information, close the window and reopen it after the analysis.

Note 2: To see procedure-level variables or constants, use the <u>Hypertext window</u>.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Problem report

(version 2.0 onwards, <u>registered version</u> only)

Project Analyzer can provide you with a **Problem report** about your project. It lists some minor but sometimes annoying things.
- Basic Files Saved As Binary - save as text to work with Project Analyzer **and** for better reliability with VB.
- Files Without Option Explicit. It is good programming practice to always declare your variables.
- Dead Procedures, i.e. subs and functions that are not in use
- Minimizable Forms Without Specified Icon
- EXE file without title and icon (this happens if you haven't made an EXE file yet)

It is recommended that you take the Problem report only after the analysis has been completed.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Cyclomatic complexity

(version 2.0 onwards, <u>registered version</u> only)

Cyclomatic complexity is a measure of the structural complexity of a procedure. It is used in software engineering. The higher the number, the more complex the procedure, and the harder it is to maintain it. Cyclomatic complexity *C* for VB procedures is calculated as follows:

C = Number of Branches + 1   *(C >= 1)*

Branches are caused by IF, SELECT CASE, DO...LOOP and WHILE...WEND statements.

"Normal" values for cyclomatic complexity range from 1 (very simple) to 9 (moderately complex). If *C* is more than 10, you may want to split the procedure.

You can see procedures with the highest complexity by selecting **View|Cyclomatic complexities**. Cyclomatic   complexities can also be viewed in the <u>file details</u> or <u>procedure details</u> window. An average cyclomatic complexity, as well as the distribution of C, is reported in the <u>Project Summary Report</u>.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Rich Text Format

(version 2.0 onwards, <u>registered version</u> only)

Project Analyzer can produce formatted <u>reports</u> and save them in a Rich Text Format (RTF) file. A number of popular word processors read RTF files, including Microsoft Word and Windows 95 WordPad.

To use RTF reports efficiently, I recommend you associate the **.rtf** file extension with your word processor. Then you will be able to run your editor directly from Project Analyzer by <u>redirecting report output</u> to **Editor** and then choosing an RTF file.

MS Word

A special option for Microsoft Word users is the **Report|RTF editor is MS Word** menu command. Use this if you have associated the **.rtf** extension with Word to optimize the reports for Word. This option will work with those versions of Word that accept English DDE commands (it will not work with the Finnish version, for example).

If this option is set, Project Analyzer optimizes the RTF reports for Word.
1) Use styles like Normal, Heading 1, Heading 2, ... You can easily create a Table of Contents for your report with Word.
2) Avoid multiple Word intances from being run if you are <u>redirecting output</u> to **Editor**.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Options

(version 2.0 onwards)

1) Show report on double click
2) Show detail window on double click

These options let you specify what happens when you double click the **file list** or the **procedure list** on the main screen. Option **Show report on double click** is for compatibility with early versions of Project Analyzer (1.0 and 1.1), and will produce the standard reports for a file or a procedure. Option **Show detail window on double click** lets you easily view some file details or procedure details.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Features in the registered version

The following features are available in the registered version only. Some of them will work in the shareware version too, provided your project is not larger than a certain limit.
- Dead code information
- Dead variable and constant information
- <u>Generate constants module</u> command
- Variable reference lists (which files reference a variable)
- <u>Cyclomatic complexity</u> for procedures
- <u>Call trees</u>
- <u>Problem report</u>
- <u>DLL report</u>
- Formatted output to an <u>RTF</u> file
- <u>Hypertext window</u>

<u>Registration information</u>

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Project summary
(version 2.1 onwards)

To get a summary of your project, use the **Report|Project summary** command. This report includes some statistics about your project. It also shows some Visual Basic limitations, like the global symbol table size of your project, and its maximum size, 64 kB.

Not all of the measures are exact, these approximated measures are marked as *(approx).*

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Generate constants module

(version 2.1 onwards, registered version only)

The use of the CONSTANT.TXT and DATACONS.TXT files in a project tends to take up a large amount of memory, because these files include so many **global const** declarations, and only a few of them are needed in one project. Including a large number of unused constants in a project will fill up the global symbol table, as well as decrease the performance of the program.

If you use Project Analyzer, you can solve the problem easily. When you are developing your application, you can include the CONSTANT.TXT and DATACONS.TXT files in your project; you don't need to cut and paste any declarations to another global module. When your project is about to be ready, just analyze it with Project Analyzer, and select **File|Generate constants module**. This command will generate a new module including only those constant declarations that are needed by your application.

After generating a new module, remove the old one (CONSTANT or DATACONS) from your project and add the new one. You're done.

Note 1: This command applies mainly to VB 3.0 projects. VB 4.0 doesn't require you to declare a vast amount of constants, but has predefined constants instead.
Note 2: This command is available only after analysis phase 2 has completed.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

DLL report

(version 2.1.07 onwards, <u>registered version</u> only)

This report lists all DLL files used by the project. All declared DLL procedures are listed, and if they are not used, they are marked as "dead".

Use this report to examine which DLLs you should distribute along with your application, and to remove unnecessary DLL declarations.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.

Nested loops

(version 3.0 onwards)

Select **View|Nested loops** to see the procedures that have nested **For..Next, Do..Loop** or **While..Wend** loops.

Generally speaking, loops are the parts in a program that take the most CPU time. When analyzing the efficiency of different algorithms, most emphasis is put on analyzing nested loops and what is inside them. With Project Analyzer, you can easily find those procedures that have nested loops.

Project Analyzer 3.0 - Helpfile generated by VB HelpWriter.