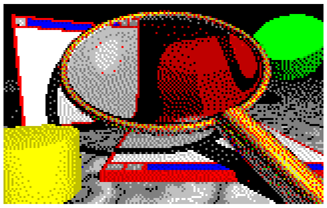
Project Analyzer 2.1



Overview of Project Analyzer

Introduction
How it works
Registration and new versions
Version history - What's new?

Menus

File

Open project...
Save analysis...
-
Archive project files...
-
Exit

View

Find procedure...
-
File details
Procedure details
Variable and constant list
-
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...
Problem report
Project summary
-

Output
    Display

        - Printer
        - File...
        - Editor...
    - Dull gray display forms
    - Use MS Word as RTF Editor
- Options
    - Show report on double click
    - Show detail window on double click

Project Analyzer was written by **Tuomas.Salste@Helsinki.Fi**, WWW page **http://www.helsinki.fi/~salste/** Information about this program and other Visual Basic tools will be available through this page.

Project Analyzer 2.1 - 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 output device defined with the **Output goes to** radio buttons.

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. An example:

```
' This sub was created by N.N.
SUB MySub (Byval x As Integer)
' 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.

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, .FRM, .BAS and .GLB files belonging to your project and the .MAK file itself are selected. 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.

Introduction

Project Analyzer is a Finnish shareware tool designed to help creating, maintaining and reporting applications developed in Visual Basic 3.0. 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 etc. in the VB project. The results can be viewed on the screen or in textual format in your favourite word processor or on the paper.

Features

Full source code analysis
- File details: procedure and control lists, file level cross-references
- Procedure details: cyclomatic complexity, lines of code, procedure level cross-references
- Declared DLL functions
- Find dead or live procedures, variables and constants
- Variable and constant declaration information, both dead and live
- Call trees

Reports for maintaining the project
- Sub and Function lists with comments for reporting purposes
- File lists with comments
- Problem report
- Variable and constant lists with dead or live variables & constants

Other powerful features
- Project file archiving with PKZIP or ARJ
- Hypertext style code viewer
- Saving the results of the analysis for future use
- FRX file view

See version history for new features only.

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** file with the Open Project command in the File menu and wait. Project Analyzer will collect information about the **.mak, .bas**, **.frm**, **.frx, .glb,** and **.txt** files **(remeber to save them in text format!)** in the background.

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 PROJTEST.MAK file. That is a pure demo project for you to see how Project Analyzer works. Just double click a procedure or a file to see the results. 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

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** file).

Project Analyzer examines this .mak 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 **.mak, .bas, .frm, .frx, .glb,** and **.txt** files belonging to your project. **Note: the files must have been saved in text format.** Project Analyzer can't currently read VB files saved in binary format, other than the .frx files. If the program finds a binary file it will simply ignore its procedures, so not much harm is done except you'll get incomplete analysis.

Warning! To get realiable dead code and dead variable & constant reports, you **must** save all your files in text format.

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!

In the registered version, if you have saved an analysis to an **.anl** file, you can reload it by opening the .anl file instead of the corresponding .mak file.

More about saving and reloading the analysis

Reports

Project Analyzer can currently produce reports in four different ways. To select one of them, click the **Output** menu or one of the **Output goes to** radio buttons. These are the ways you can select:

1. To the display
This is the default. The output goes to a special Display window where you can read the text. There are 3 of these Display windows available. You can edit and copy the text onto the clipboard.

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

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

4. To a text editor
If you want, Project Analyzer can, after saving its output to a file, run an editor (like Notepad, Write, or MS Word) so you can quickly see the output and edit it. To run your text editor from Project Analyzer you should associate the file name extension (**.txt**, **.wri** or **.rtf**) with your editor using the Windows File Manager. (by default, .txt files are associated with Notepad and .wri files with Write).

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 display 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 printer, text file or RTF file option, which allow reports of any length.

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 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.

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 the future versions for free. This requires an email address. The registered users receive further updates and information about Project Analyzer automatically by email. I will send the files to you uuencoded by email.

Price

The price (valid at least the year 1995) for registering version 2.1 is:

USD	50	USA
FIM	200	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 valid provided there are no drastical changes in the exchange rates.

GBP	35	Great Britain
DEM	75	Germany
SEK	400	Sweden
NOK	350	Norway

The price is a bit cheaper in Finnish marks, because I don't have to bear the exchange costs then.

The source code is also available for an extra fee.

Payment method

1. Cash: The standard way of paying the registration fee is sending cash in a letter. That's the cheapest way outside Finland.
If you want a more secure payment method, you must pay all the additional transfer costs.
2. Checks: If you would like to use a check, please add $15 to the price, because I will have to pay that for cashing the check at my bank.
3. Registered mail: You can register your letter at your local post office. That will typically cost a few

dollars.
4. Electronic transfer: The most secure payment method is an electronic funds transfer (SWIFT transfer). You must consult your local bank about that. That will typically cost about $15 to the sender, but it has the advantage that you can send Finnish marks using it, and the price is a bit lower in Finnish marks. SWIFT is fast too. Please ask me about my SWIFT account number.
5. Bank transfer in Finland: This is the best way (I'd say the only one) to pay in Finland. Please ask me about my account number.

My address is:
Tuomas Salste
Mäkitorpantie 29-31 A 12
00640   HELSINKI
FINLAND

I can send a receipt if you request it.

Other utilities:

- 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
- Status - a status bar to tell you the time and free space on your hard disks

See my WWW page (**http://www.helsinki.fi/~salste/vb.html**) or mail to me for more information!

Hypertext

(version 1.1 onwards, registered version 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 underlined, you can jump to the declarations section where it was declared.

Color key:

Green	Procedure in this file
Violet	Procedure in another file
Dark Blue	Module-level variable/constant
Light Blue	Global variable/constant

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

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.

Note: This feature is available in the registered version only. In the unregistered version you can only use it with the **projtest.mak** that comes with Project Analyzer.

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 Variables and constants window. 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 output 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 Variables and constants window.

Saving and Reloading the Analysis

(version 1.1 onwards, registered version only)

Once you have completed the analysis phases you may want to save the results in order to quickly use Project analyzer again. You can save the results in an **.anl** file that will be named after your **.mak** file and located in the same directory.

When you open an **.anl** file with the **File|Open project** command, Project Analyzer will check if the files that belong to the project have changed. If not, the analysis is quickly in your use again. And if a file has changed, Project Analyzer will analyze it from scratch, but the unchanged files will need no operation. This will speed up the use of Project Analyzer.

If you notice the **.anl** file is badly out of date (e.g. you have done extensive reorganization on your code by moving functions to other files etc.) you can always do whole the analysis from scratch by opening the **.mak** file instead of the **.anl** file.

Note: You cannot read files saved in earlier versions of Project Analyzer, because those files don't contain all the information that this version can analyze. You must analyze your project again.

Collecting information

Version History

Version 2.1 (September 1995):
- Project summary report
- Generate constants module command
- Modifications to look better with Windows 95
- Enhancements in the hypertext window
- About 25% faster analysis (appears in the second phase) due to improving PROJECT.DLL

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, registered version 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

These trees are available using the **View|Procedure call tree** and **View|File call tree** menu commands.

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 the demo project **projtest.mak.**

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

Note 3: You can see the procedural dependencies in hypertext or list form too.

If you don't like the colors I've chosen for the Display forms, you can set them to gray using this option.

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:
- FRM files (forms files)
- FRX files
- BAS, TXT and GLB files (modules)
- DLL's

Basic files (FRM, BAS)

- 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 files are not analyzed by Project Analyzer.

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.

Variables and constants window

Project Analyzer can show a list of all global and module-level definitions either as a textual report 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: Const and Global Const definitions
- Show variables: Dim, Global and Static definitions
- Show global definitions: Global and Global Const definitions
- Show module-level definitions: Dim, Static and Const definitions
- Sort by file
- Sort by name

To see the actual definition of a variable or a constant, press the **View declaration** button. That will bring up the Hypertext window with the corresponding (declarations) section. To view a textual listing, press the **Report** button. And to see which files reference the selected variable or constant, select the **View references** command.

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: Project Analyzer doesn't currently collect any procedure-level variables or constants.

Problem report

(version 2.0 onwards, registered version 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.

Cyclomatic complexity

(version 2.0 onwards, registered version 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 should think about splittting the procedure.

Cyclomatic complexity can be viewed in the file details or procedure details window.

Rich Text Format

(version 2.0 onwards, registered version only)

Project Analyzer can produce formatted reports and save them in a Rich Text Format (RTF) file. A number of popular word processors read RTF files.

To use RTF reports efficiently, I recommend you associate the **.rtf** file extension with your word processor using the Windows File Manager. The you will be able to run your editor directly from Project Analyzer by redirecting output to **Editor** and then choosing and RTF file.

A special option for Microsoft Word users is the **Report|Use MS Word as RTF viewer** menu command. Use this after you have associated the **.rtf** extension with Word to avoid multiple Word intances from being run. This will work with those versions of Word that accept English DDE commands (it will not work with the Finnish version, for example).

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 2.1 - 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 with the included sample project PROJTEST.MAK too.
- Dead code information
- Dead variable and constant information
- Generate constants module command
- Variable reference lists (which files reference a variable)
- Cyclomatic complexity for procedures
- Hypertext code window
- Call trees
- Problem report
- Formatted output to an RTF file
- Saving the results in an .ANL file to be loaded later for increased speed

Registration information

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 2.1 - 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: This command is available only after analysis phase 2 has completed.