

Code.Print for Visual Basic Help

Table of Contents

General

HOW Code.Print for Visual Basic PRINTS

Printing from Project MAK Files

Printing Single Files

Printing Single Routines

Configuration and Options

File Menu Commands

Visual Basic 1.0 and 2.0 Compatibility

General

Code.Print for Visual Basic is designed to provide the developer with an affordable, capable tool to produce easily maintainable presentation quality source code documentation. Code.Print for Visual Basic provides the user with a rich set of printing options:

Header/Footer Options:

- Date and Time
- Project Name
- File Name
- Header/Footer Notes
- Routine Name
- Page Numbers
- Continuous page numbering
- Starting Page
- Three header lines for additional notes
- Three footer lines for notes

Page and Output Options:

- Line Spacing
- Top, Bottom, Left and Right Margin Control
- Sort Output by Sub-Routine
- Sort File by Sub-Routine
- Serialize Backups
- Page Breaks on New Routines
- Fit Subroutine on Page
- Background Printing
- Bold Routine Name
- Subroutine Index

Fonts:

Individual fonts, styles, and sizes can be designated for Header/Footer, Procedure Title, Comment, and Code lines. Any font available to the printer is available for your selection.

How Code.Print Prints

Code.Print for Visual Basic prints directly from Visual Basic 2.0 .FRM or .BAS files that have been saved in ASCII format. Code.Print can also print files saved as text with a TXT extension. Visual Basic 1.0 .FRM and .BAS files cannot be printed directly as these files are in a proprietary binary format and thus are not usable for printing purposes.

Note: Although you can create text files by hand using the **Save Text** command, and convert binary files to ASCII by using the **Save As...** command, a utility is available from Caladonia Systems, TexMake, that automates these processes.

You can also print files created with other editors provided they are in TEXT format. Be sure to set the **sort options** off and set the **comment font** attributes the same as your **code font** settings if you want to print non-program source code text files. (See configuration)

Printing from Project MAK Files

Code.Print for Visual Basic is capable of reading project MAK files. By selecting a MAK file in the **File** list box, Code.Print for Visual Basic builds a sorted list of all forms and modules and proceeds to find and print all associated code files available. This enables the user to print an entire application with one command.

To print from a project make file, select the MAK file you want in the FILE list box. (You can also type the name of the file in the text box directly. Be sure to include the .MAK extension.) Select **FILE PRINT (Ctrl-P)** to start printing.

When Code.Print for Visual Basic prints from a MAK file, page numbers increment continuously until the end of the print job if **Continuous** is checked in the Header/Footer Options group. Otherwise page numbers reset to the starting page with each new text file. Page number references in the index reflect whatever page numbering method was used.

Visual Basic 1.0 Tip:

Although Visual Basic 1.0 allows you to have a form and module with the same name in a project (ie. SAMPLE.FRM and SAMPLE.BAS,) only one of the files can have a text code file (ie. SAMPLE.TXT) Code.Print for Visual Basic will print the text file twice, once for each occurrence of the root name in the project make file. To avoid this confusion, give each form and module file in a project a unique name.

Printing Single Files

To print a single file, select the file you want in the FILE list box.

Select FILE PRINT (Ctrl-P) to start printing.

Printing Single Routines

To print a single procedure within a code file, select the file you want in the **FILE** list box.

Code.Print will scan the file and make a sorted list of the procedures it finds. This list appears in the **PROCEDURES** box. The default procedure is **[All]**.

Select the procedure you wish to print and then select **FILE PRINT** (Ctrl-P) to start printing. Select **[Declarations]** to print only the declarations section of a TXT file.

Code.Print for Visual Basic does not print an index when printing single routines.

Configuration Settings

Overview:

CURRENT SETTINGS and **STARTUP CONFIGURATION** windows are identical in appearance and use. The two differ in that startup configuration settings are saved to disk and are set when the program loads.

CURRENT SETTINGS enables you to change any settings during the current session. They remain in effect until changed or the program is exited.

The CURRENT SETTINGS window also contains a command button labeled **DEFAULT**. Pressing DEFAULT resets all options to pre-set values.

Saving Configurations and Settings:

If you are in the CURRENT SETTINGS window, settings are saved as they are changed. The OK button restores the Main Window.

The STARTUP CONFIGURATION window has SAVE and CANCEL buttons. Pressing SAVE will write the configuration settings to disk. Pressing CANCEL will exit the window without saving the changes.

Configuration Topics:

[Header/Footer Options](#)

[Fonts](#)

[Header and Footer Notes](#)

[Page and Output Setup](#)

[Font Setup](#)

Header/Footer Options

Date and Time - Check this box to print Date and Time.

Project Name - Check this box to print Project Name.

Project Name Text - If Project Name is checked you can enter text into the Project Name Text box.

File Name - Check this box to print TXT file name.

Include Path - Check this box to print the full path name of the file.

Header/Footer Notes - Check this box to print header and footer notes, if they exist.

Header/Footer Border Lines - Check this box to print horizontal lines that separate the header and footer from the source code print area.

Print Routine Name - Check this box to print the name of the routine on the title line.

Page Numbers - Check this box to print page numbers.

Continuous - Check this box to print page numbers without resetting to 1 at the start of a new file. This option works only when printing from a MAK file. Starting Page - Enter the page number to start (default 1).

Date, Project, Filename, Header Notes, and Print Routine Name can be limited to print on the First Page Only for each file. Check the **Page 1 Only** box next to the option you want to limit.

Fonts

Individual fonts, styles, and sizes can be designated for Header/Footer, Procedure Title, Comment, and Code lines. Any font available to the printer is available for your selection but must first be assigned to the Code.Print font set. (See Font Setup.)

To change a font, press the FONT list button. Select the desired font by double clicking on it.

Font Sizes:

To select a font size, press the SIZE list button for the font you want to change. Select the size you want.

Note:

If the font size selected is not available Code.Print for Visual Basic will select the nearest usable size.

Font Styles:

To select a font style check the appropriate style box. There are style check boxes for BOLD, ITALIC, and UNDERLINE. Normal style is the default when no styles are checked.

Header and Footer Notes

Header Note Lines - To enter/edit header note lines press the EDIT HEADER/FOOTER command button. Code.Print for Visual Basic will display the current header lines. Edit as you need and press OK when done.

Footer Note Lines - To enter/edit footer note lines press the EDIT HEADER/FOOTER command button. Code.Print for Visual Basic will display the current footer lines. Edit as you need and press OK when done.

Page and Output Setup

To change the Page and Output Setup press the PAGE AND OUTPUT SETUP button on the CURRENT SETTINGS or STARTUP CONFIGURATION form.

Code.Print for Visual Basic will display the Page and Output Options form. Press OK when finished.

Margins - Enter the size of the margins (Select in or cm)

Sort Output - Check this box to print the routines in each file in alphabetical order.

Sort File - Check this box to sort the TXT file by procedure and save it to disk. (Useful for setting up the file for uses by editors, etc. Code.Print for Visual Basic preserves the declarations section at the top of the file. This process sorts TXT files only. FRM and BAS files cannot be sorted.

Serialize Backups - When Code.Print for Visual Basic sorts a TXT file it creates a backup of the original TXT file. If **SERIALIZE BACKUPS** is checked, Code.Print for Visual Basic copies the original file to a file with the next logical serialized extension. For example, with a file called SAMPLE.TXT, say the last backup created is SAMPLE.V05. The next backup file that Code.Print for Visual Basic makes will be SAMPLE.V06. Versions go from V00 to V99 and then start over. If SERIALIZE BACKUPS is not checked the backup file always has the extension of BAK.

Page Break on New Subroutine - Check this box to start a new page for each new procedure encountered.

Fit Subroutine on Page - Check this box to insure that any routine that is larger then can be printed on the current page will be printed starting at the top of the next page. This option is not available if **Page Break on New Subroutine** is checked.

Line Spacing - Enter the line spacing you want (Default 1)

Print Subroutine Index - Check this box to print an index of the procedures printed during a print job. The index lists the procedure name, file, and page number where the procedure was printed. When printing from a MAK file, the index list is alphabetized based on the entire list of procedures encountered during the print run. Indexes are not available when printing single procedures.

Print While Processing - Check this box to release output to the Print Manager as soon as each TXT file is completed. This option will allow you to split the output of a MAK file print job into several individual print files rather than one very large print file. It will also allow you to print in background while Code.Print processes the files.

Bold Routine Name - Check this box to highlight the name of each subroutine or function in the printout. Make sure your code font is not set to bold, otherwise you will see no difference in the output.

Font Setup

To improve performance, Code.Print for Visual Basic uses a sub-set of the total fonts available to the system. Since normal operation of Code.Print for Visual Basic requires only a few fonts be available at any given moment a substantial improvement in performance can be realized by referencing this small sub-set rather than the total fonts available. This is especially true in cases where the total number of fonts available is very large.

ASSIGNING FONTS:

To assign fonts to the Code.Print font set, select OPTIONS FONT SETUP (Ctrl-F).

The Font Setup window has two list boxes. On the left is a list of unassigned fonts available. On the right is a list of the current assigned Code.Print fonts. Double clicking on a font (or highlighting and pressing ENTER) will move the font from one list to the other. When you are finished with your selections, press SAVE to set the Code.Print font set to the assigned list. The CANCEL button exits the Font Setup window without changing the current assigned font list.

If you do not assign any fonts to the assigned font list Code.Print will set the default font to Courier.

File Menu Commands

Print - Select this option to start a print job. You can also use the quick key Ctrl-P.

Print Setup - Select this option to access the Print Setup Dialog Box

Exit - Select this option to exit the program. You can also use the quick key Ctrl-X or select CLOSE from the control box menu on the Code.Print for Visual Basic main window.

Visual Basic 1.0 and 2.0 Compatibility

Code.Print for Visual basic can automatically determine which version of Visual Basic created a selected MAK file.

If you are processing Visual basic 1.0 files, Code.Print will search for corresponding TXT files for each form and module in the project. Code.Print cannot process Visual Basic 1.0 files directly because they are saved in a binary format that Code.Print cannot read.

If you are processing Visual Basic 2.0 files, Code.Print will try to process the project files directly. However, if the FRM or BAS files are not saved in ASCII text format then Code.Print will look for a corresponding TXT file. If the TXT file is not found Code.Print will display a message indicating that it could not process the file.

It is generally a good idea to save you projects in ASCII format. A conversion utility, Caladonia Systems TexMake, will automatically convert projects from binary to ASCII and back again.

