

WinAsm Help Contents

Introduction

Requirements

Features

Operation/Usage

Controls

Menu

SmartButtons

PopUp Menu

Indicators/Status

About Insomnia Software

Introduction

WinAsm is a Windows based development environment for assembly code development. It is based on DOS based assembly code development tools.

WinAsm provides a Multiple Document Interface (MDI) for editing source files. This allows the user to work with several source files and a make file concurrently.

WinAsm provides a user configurable control for Assembly, Linking, Running, Make, and UnMake. This allows the user to use any set of DOS based assembly tools. The user must provide an assembler, linker, and make utility.

The status lights to indicate the status of the current file. They indicate if a corresponding object file, executable, map file, cross reference file, and make file exist for the active file in the current directory.

The Smart Buttons provides quick access to the WinAsm specific functions.

<u>Requirements</u>	How much machine does it need ?
<u>Features</u>	What WinAsm can do for you
<u>Operation/Usage</u>	General description of how to use WinAsm

Requirements

WinAsm itself will run on any system running Windows 3.1 or higher with 2 megs of RAM.

In order to assemble or link files a system must be operating in Enhanced mode with a minimum of 4 megs of RAM.

Features

WinAsm is a Windows Development Environment for Assembly code. It has the following features.

MDI interface to simplify operation on multiple files

Status Lights to indicate current status of corresponding files.

Smart Buttons for assembly operations and output file display.

Status bar for system information.

PopUp Menu for quick edits and help file searches

User configurable Usage of (DOS) Assembler, (DOS) Linker, and (DOS) Make Utility.

User configurable Help Files.

Operation/Usage

Background

WinAsm a MDI Windows text editor with several controls that execute DOS command line functions. These command line functions are intended to simplify the development of Assembly code using DOS based tools such as Borland's Turbo Assembler, or Microsoft Masm.

Full operation of WinAsm requires that the USER supply the DOS based Assembly tools.

Since WinAsm does execute DOS calls, it must operate according to a PIF file. When Windows is directed to execute a DOS call without a PIF file, it uses '_default.pif' in the windows subdirectory. In order to fully utilize the functions in WinAsm, the user must modify the '_default.pif' file in the windows subdirectory.

User Configuration/Installation

At this point we can assume you have created a subdirectory for WinAsm and have installed the icon in a program group. There are additionally 2 more steps to perform to fully install WinAsm.

1) Configuration of WinAsm - Select the WinAsm button. At the configuration dialog box, you must enter the DOS command line of the Assembler, and Linker. They default to Borland Turbo Assembler. If a Windows help exists (one does in Turbo Assembler 4.0), the name must be entered in the help section.

2) Modify _default.pif - Using the PIF Editor (in the Main Program Group), load the _default.pif file in the windows subdirectory. Select windowed operation and XMS memory of -1, EMS memory of 0, and Exclusive Operation, and Close Window on Exit. Save the file back to disk. Installation is complete

Controls

WinAsm has both the title bar menu and a left side SmartButtons. The top menus contains functions applicable to the process of file operation and text editing. The SmartButtons contains functions that are unique to WinAsm (with the exception of Open and New). The PopUp Menu is available by clicking the right mouse button while in the active child window.

Menu

<u>File</u>	File operations
<u>Edit</u>	Clipboard operations
<u>Character</u>	Alignment and font operations
<u>Window</u>	MDI window operations
<u>Help</u>	WinAsm Help

SmartButtons WinAsm Specific operations

PopUp Menu Mouse operations

Menu

The menu contains a number of functions used to create, edit, and save files within a MDI editor.
The Main menus are:

<u>File</u>	File operations
<u>Edit</u>	Clipboard operations
<u>Character</u>	Alignment and font operations
<u>Window</u>	MDI window operations
<u>Help</u>	WinAsm Help

Menu : File

New

Creates an unnamed child window in WinAsm.

Open

Displays the common dialog for opening a file.

Close

Destroys the current active child window. If the active child window is modified, prompts for file save.

Save

Saves the contents of the active child window. If the active child window is unnamed, the SaveAs dialog is displayed.

SaveAs

Displays the common dialog for saving a file.

Print

Prints the contents of the active child window in WinAsm.

Print Setup

Displays the common dialog for printer setup and selection.

Exit

Prompts for file saves if modified, and exits WinAsm.

Menu : Edit

Cut

Cuts the current selection in the active child window to the system clipboard. If there is no selection this option is disabled.

Copy

Copies the current selection in the active child window to the system clipboard. If there is no selection this option is disabled.

Paste

Pastes the clipboard contents to the cursor location in the active child window. If there is a selection, the selection is replaced by the clipboard contents.

Delete

Clears the current selection in the active child window.

Select All

Selects all of the contents within the active child window.

Menu : Character

Left

Left alignment indicates that the text is left aligned within the active child window. This is the default setting.

Right

Right alignment indicates that the text is right aligned within the active child window.

Center

Center alignment indicates that the text is centered within the active child window.

Word Wrap

Wraps the line to the next lower line if the length exceeds what can be displayed in the active child window.

Font

Displays the common dialog for selecting a font. This font will apply to the active child window, and any new windows created or opened after the font selection.

Menu : Window

Tile Vert

Displays the child windows in a non-overlapping arrangement along vertical divisions.

Tile Horiz

Displays the child windows in a non-overlapping arrangement along horizontal divisions.

Cascade

Displays the child windows in an overlapping arrangement, with active child on top with all title bars accessible.

Arrange Icons

Arranges the minimized windows within the workspace.

File List

Lists all of the edit windows available in the workspace. By selecting the filename in the filelist, this becomes the active and visible window.

Menu : Help

WinAsm Help

Displays the Main Contents for this file.

Asm Help

Displays the Main Contents of the Assembler Help file. This file is not included with WinAsm, and must be specified by the user. If an Assembler help file is not available, this selection does nothing. Although this help control is intended to be a reference to the Assembler usage, it may reference any help file.

X86 Help

Displays the Main Contents of the X86 Help file. This file is not included with WinAsm, and must be specified by the user. If an X86 help file is not specified, this selection does nothing. Although this help control is intended to be a reference to the X86 processor set of opcodes and instructions, it may reference any help file.

About

Displays an About screen for WinAsm.

SmartButtons

New

Creates an unnamed child window in WinAsm.

Open

Displays the common dialog for opening a file.

Asm

Assembles the file corresponding to the active child window. The Asm button enabled only if the active child window has the '.asm' file extension, and is unmodified. When debugging an application, the '_default.pif' file may be modified to NOT close on exit. This will allow the DOS Window to remain open after it is done executing. For more information see Configure.

Link

Links the object file corresponding to the active child window. The Link button enabled only if the active child window has a corresponding object file in the same directory. When debugging an application, the '_default.pif' file may be modified to NOT close on exit. This will allow the DOS Window to remain open after it is done executing. For more information see Configure.

Make

Executes the make file in the active directory.

Run

Runs the executable corresponding to the active child window. The Run button is enabled only if this file exists in the active directory. If the application is a DOS application, it will execute according to the '_default.pif' file in the Windows subdirectory. When debugging an application, the '_default.pif' file may be modified to NOT close on exit. This will allow the DOS Window to remain open after it is done executing.

Unmake

Deletes the file types specified in the configuration associated with the active child window.

List

Creates a new child window in the workspace and load the list file corresponding to the active child window. The List button is enabled only if the file exists.

Map

Creates a new child window in the workspace and load the map file corresponding to the active child window. The Map button is enabled only if the file exists.

Configure

Displays a configuration dialog. This dialog allows the user to select options for the assembler, the linker, make, unmake, and help files. The command line input specifies the name of the DOS executable that performs that function. Minimize on use specifies that the application runs as an icon rather than a window. In order for minimize on use or windowed operation to operate correctly, the file '_default.pif' in the Windows subdirectory must be modified to allow Windowed operation. The cross reference and list file options for the Assembler will cause the assembler to create the corresponding files (if the assembler is Tasm/Masm compatible). The filetypes under Unmake allow the user to specify which file types to delete on Unmake. The filename input under the help section is required for WinAsm to support the help functions.

About

Displays an About screen for WinAsm.

PopUp Menu

The PopUp menu appears when the user clicks the right mouse button while on an active child window. The menu provides the following functions:

Cut

Cuts the current selection in the active child window to the system clipboard. If there is no selection this option is disabled.

Copy

Copies the current selection in the active child window to the system clipboard. If there is no selection this option is disabled.

Paste

Pastes the clipboard contents to the cursor location in the active child window. If there is a selection, the selection is replaced by the clipboard contents.

Asm Topic Search

Displays the results of searching the keywords of the Asm Help file. This file is not included with WinAsm, and must be specified by the user. If an Assembler help file is not available, this selection does nothing.

X86 Topic Search

Displays the results of searching the keywords of the X86 Help file. It file defaults to the file included with WinAsm, and may be specified by the user. If an X86 help file is not specified, this selection does nothing.

Indicators/Status

WinAsm has several different status indicators. These include the Title of the Active Child Window, the Status bar across the bottom of main window, and the status lights on the lower left side of the main window.

<u>Title Bar</u>	Filename and Modified Status
<u>Status Bar</u>	System Status Display
<u>Status Lights</u>	Assembly File Status indicators

Title Bar

The Title bar of the child windows displays the full filename of the contents. This filename is used to determine the state of WinAsm controls and status lights. The filename is followed by a 'modified ' if the contents of the window have been modified since last saved.

Status Bar

The Status Bar on the bottom of the window displays status relevant to the operation of WinAsm. The information shown is:

Last Command

The display window on the left end of the status bar holds the last DOS command issued by WinAsm.

Active File

The display window in the center of the status bar displays the filename of the current active file.

System Time

The display window on the right end of the status bar displays the current system time.

Status Lights

The Status Lights on the lower left side of the WinAsm window indicate the file status of the current directory and active file. These the status lights are:

Obj

Green if an object file for the the active child window exists in the current directory. Red for all other cases.

Exe

Green if an executable file for the the active child window exists in the current directory. Red for all other cases.

Lst

Green if an list file for the the active child window exists in the current directory. Red for all other cases.

XRF

Green if an cross reference file for the the active child window exists in the current directory. Red for all other cases.

Map

Green if an map file for the the active child window exists in the current directory. Red for all other cases.

Make

Green if a make file exists in the current directory. Red for all other cases.

About Insomnia Software

Insomnia Software specializes in quality software for obscure (and sometimes obtuse) applications.

Development System

I am using a AMD486dx66 with 8 megs of RAM. I write all of my code in the following packages:

Borland Pascal 7.0
Borland Turbo Assembler 4.0
Borland Delphi (version FT5)

Software Charter

As the sole employee of Insomnia Software, I am able to follow any whimsical path in the exploration of software development. This means that my software is far from the beaten path, and may not have any useful application. Most of the software packages were written as tools or examples to simplify development in other areas. For example, the VLI packages were written to simplify some Crypto software I was considering. WinAsm was written to simplify the writing of VLIx32.

Products

The software packages available from Insomnia Software are:

<u>IS Screens</u>	Window 3.1 Screen Saver Collection
<u>VLI64</u>	Windows Crypto Programming Tools (16 bit code/64 bit integers)
<u>VLIx32</u>	Windows Crypto Programming Tools (32 bit code/64/128/256/512 integers)
WinAsm	Windows Development Environment for Assembly Coding

If you are interested in distributing these packages or licensing the source code, I can be contacted at TomWWolf@aol.com.

About IS Screens

A collection of Screen Savers for Windows 3.1 or higher. These savers are built using a Borland Pascal 7.0 Screen Saver Object. This object supports full password protection, full profile support, and hidden screens. The screen savers are freeware. The source code and Saver Object is buyware. The savers include:

Balls

Balls is a screen saver that bounces a number of colored balls around the screen, leaving a trail behind. The number, size, and colors of the balls are user configurable. The balls change path when they hit the sides of the screen, or occasionally a random direction shift.

Worms

Worms is a screen saver that has a number of colored worms that eat up the screen. The worms move randomly, but stay within the sides of the screen. The number, size, and colors of the worms are user configurable.

Julia

Julia is a screen saver that draws a Julia set on a black background. It is a visually quiet screen saver. It is user configurable.

Slide Show

Slide Show is a screen saver that periodically displays a bitmap to the screen. It will display 256 color bitmaps. It is user configurable.

Blackout

Blackout is a screen saver that blacks out the screen. It is just like the blackout included with Windows 3.1, except it supports passwords.

About VLI64 Toolkit

A Programmers collection of 42 mathematical functions applicable to large integers. It includes 42 functions and procedures for 64 bit integers. These functions include logical tests, mathematical, bit manipulation, and type conversion. Most of these functions are written in 16-bit assembly. This collection of functions is designed to simplify the task of cryptography research for the casual cypherpunk. The functions are accessible through a Borland Pascal 7.0 unit or a Windows DLL. The package is freeware. The source code is Buyware.

VLITest

VLI**Test** allows the user to functionally test and benchmark each of the 42 functions.

VLIEval

VLI**Eval** allows the user to enter 64 bit hexadecimal numbers and apply any of the 42 VLI64 functions, and observe the results.

About VLix32 Toolkit

A Programmers collection of mathematical functions applicable to large integers. It includes the basic 42 functions for 64, 128, 256, and 512 bit numbers. Many of the core functions will be written 32 bit assembly. This collection of functions is designed to simplify the task of cryptography research for the casual cypherpunk. The functions are accessible through a Borland Pascal 7.0 unit, a Borland Delphi unit, or a Windows DLL. The package is freeware. The source code is Buyware.

VLITest2

VLITest2 allows the user to functionally test and benchmark each of the 42 functions.

VLICalc

VLICalc is a calculator that uses the VLI functions. Operates in decimal or hexadecimal.

