Windows Disassembler 1.3

Introduction

Windows Disassembler disassembles Windows 3.x programs and Dynamic Link Libraries. It automatically displays assembly language source code within the window and allows you to optionally create assembly language files.

Commands

The Display Window

The display window displays the assembly language code of a segment from the program which you have opened, excluding certain directives (i.e., **TITLE**, **.MODEL**, **.286**, etc.). At the top of the window is the following information:

Segment number
The chronological number of the current segment. (This number is based on the order in which segments are placed in the executable file)

Segment offset
A hexadecimal value equal to the segment's offset relative to the beginning of the file

Segment size
A hexadecimal value equal to the number of bytes in the segment.

Bytes in the program can be changed from data declarations to instructions and vice versa with the Set Bytes command. The offset of each instruction can be displayed with the Address Offsets command. Only one segment can be displayed at a time. The **Segments** command allows you to choose which segment you want displayed. To create an assembly language file, use the Save Text As command. The **Far Call Names** command toggles between showing the Windows API function call names and the actual relocation bytes.

Creating and Assembling Assembly Language Files

You can create an assembly language file for the program that you've opened by using the Save Text As command. Note, however, that the source code created by *Windows Disassembler* cannot be reassembled without some editing. Also, you will need the resource files (which can be extracted using Borland's *Resource Workshop*), and a definition file (which can be created with the aid of a program file header utility such as *TDUMP* or *EXEHDR*). You will then need to make the following alterations:

1. Change the model type in the **.MODEL** directive from **MEDIUM** to the correct type, unless it is a medium model program.
2. Supply all relocatable address references (i.e., **OFFSET ABOUTDLG**). These are listed in the file header listing.
3. Specify the entry point for the **END** directive (also given in the file header listing).
4. Specify (or rename if you wish) exported procedures. For example, **Procedure*n*** could be renamed as **AboutDlg** if it is exported and they are equivalent to each other (i.e., if ***n*** is the ordinal number of **AboutDlg** given in the file header listing). In some cases you will have to provide the **PROC** directive for an exported procedure if the procedure is not in the relocation table.
5. Make the data segment accessible to all modules via include files.

File Commands

Open

Opens a Windows executable file or dynamic link library. The default extension is **.EXE**.

Save Text As

Creates an assembly language source code file.

If the **Create Separate Files For Each Segment** option is checked each file's name will be given an integer file name extension equal to the segment's number (if you specify an extension, it will be ingnored).

If the **Create One File Containing All Segments** option is checked one file will be created containing the whole program. Select this option only if the program is relatively small, i.e., has only two or three segments. The default file name extension is **.ASM**.

If **Save Current Segment Only** is checked, a file is created containing only the segment that is currently being displayed in the window. The default file name extension is **.ASM**. This option preserves the byte settings made for the segment.

NOTES: If a file having the specified name already exists, it will automatically be overwritten. *Saving files erases all byte settings* (see Set Bytes) made in the current segment unless you select the **Save Current Segment Only** option.

Edit Commands

Set Bytes
Controls the way *Windows Disassembler* translates a (set of) byte(s). If **Byte Data** is selected, all bytes within the specified range are translated as byte declarations. If **Instruction** is selected, all bytes within the given range are translated as machine instructions. The **Labeled Instruction** option allows you to give labels to all instructions within the specified range (note that byte declarations are automatically given labels). Hexadecimal numbers are required for the range values.

View Commands

Go To
Goes to a specified address in the current segment (requires a hexadecimal value).

Address Offsets
Shows the offset of each instruction in the far left-hand column.

Far Call Names
Toggles between showing the Windows function call names and the actual relocation bytes.

Segment
Goes to the segment that you specify.   Note that all byte format changes which you have made will be lost if you change segments.   You can save your byte settings in an assembly language file before you change segments but you cannot retrieve the changes again (except by resetting the bytes manually)

Scrolling the Screen

PAGE UP KEY
Scrolls the window up one page.
PAGE DOWN KEY
Scrolls the window down one page.
UP ARROW KEY
Scrolls the window up one line.
DOWN ARROW KEY
Scrolls the window down one line.