

Index

[New Features in Soft-ICE/W Version 1.1](#)

[Readme Information](#)

[Soft-ICE/W Diskette](#)

New Features in Soft-ICE/W Version 1.1

Support for Windows 3.1

Support For Microsoft C 7.0

New Commands

HWND Display Window Handles

CLASS Display Window Classes

VM Display Virtual Machine Information

BPRW Range breakpoint on Windows module or selector

Loading Exported symbols from DLL's using EXP keyword

Enhancements to VIDMODE

Enhancements to WLDR

.SYM File Support

Automatic Symbol Table Switching

Improved support for Debugging Windows Drivers at source level

Improved support for Debugging VXD's at source level

Improved support for Debugging Windows Apps and DLL's

Miscellaneous changes

HWND

SYNTAX: **HWND** [Window-Level] [Task-Name]

For each window the following information is displayed

Window handle

Queue handle

Owner - this is the task name of the queue owner

Class name

Window procedure address

If no parameters are provided, all windows will be displayed.

If a Window-Level is provided only windows at that level will be displayed.

If a Task-Name is provided, only windows belonging to that task will be displayed.

CLASS

SYNTAX: **CLASS** [Module-Name]

For each window class the following information is displayed:

Class handle

Class name

Module-Name of class owner

Window procedure address

If no parameters are specified, all window classes will be displayed. If a module name is specified, only classes belonging to that module will be displayed.

VM

SYNTAX: **VM** [VM-ID]

For each virtual machine the following information is displayed:

VM Handle

VM Status bits

VM high linear address

VM-ID

VM's Client Register Structure address

If a VM-ID is specified the Client Register Structure for that virtual machine is displayed. The Client Register Structure can have both a protected mode section and a V86 section. Which ever line is on top is the last mode the VM was executing in. This can be used to determine the current cs:ip in both protected and V86 mode for any virtual machine. These structures are valid whenever executing in 32 bit VxD mode.

BPRW

SYNTAX: **BPRW** Module-Name | Selector [R|W|RW|T|TW]

This command is a shorthand method for specifying range break points on Windows applications or DLL's. If a selector is specified, a range break point will be set on the entire segment. If a Module Name is specified, a range break point will be set on every code segment belonging to that module. This is most useful for the following two purposes:

By specifying a module name with the T verb an entire Windows app or DLL can be backtraced.

By specifying just a module name, Soft-ICE/W will break the next time any code is executed belonging to that module.

EXP

EXPORTS can now be loaded from any Windows application or DLL not just User, Gdi and Kernel. Those three are automatically loaded by Soft-ICE/W.

To load additional exports, a new keyword has been defined for WINICE.DAT.

EXP=File Name

For example:

EXP=C:\WINDOWS\SYSTEM\VGA.DRV

This would cause Soft-ICE/W to load all exported symbols contained in the VGA display driver.

VIDMODE

A new VIDMODE utility is provided. VIDMODE has a new option of trapping I/O that takes place in the virtual display driver. For this option the switch to text mode can be trapped, the switch back to graphics mode can be trapped, or both can be trapped. The recommended way of running VIDMODE is as follows:

First run VIDMODE with no boxes checked. If this doesn't work go to the next step.

If Soft-ICE does not display its text mode screen correctly check the TEXT box in the VDD trapping section. If Soft-ICE does not restore the windows screen correctly check the GRAPHICS box in the VDD trapping section. If Soft-ICE does neither correctly, check both boxes in the VDD trapping section.

WARNING: When either the GRAPHICS or TEXT boxes are checked your system may hang. So before checking either of these boxes, make sure you don't have any other Windows apps running.

Two new command line utilities are provided along with VIDMODE. These utilities should only be used by people who have an in depth knowledge of a particular video board, most likely people developing drivers for that board. They can be used to create a VID file that Soft-ICE can use in the case where the VIDMODE utility does not function correctly. The utilities are described below:

V2A.EXE

This program will take the WINICE.VID file created by VIDMODE and translate it into an ASCII file named WINICE.ASC. This file can then be edited with a text editor.

A2V.EXE

This program will take the WINICE.ASC file and translate it into a VID file named WINICE.VID. This file can then be used by Soft-ICE/W.

WLDR

The Soft-ICE/W Windows program loader has been enhanced. Some of the features provided in the new WLDR are listed below:

The user interface has been redesigned.

On-line help has been added.

The last five programs that have been loaded by WLDR are now saved and can be selected with a pull down menu pick.

Windows 3.1 style file selection dialog boxes have been added.

A command line can now be passed to WLDR.

WLDR can now create a program manager icon for frequently loaded applications.

Symbols can now be loaded without loading the application or DLL

NOTE: The new WLDR requires the COMMDLG.DLL dynamic link library.

Window-Level

Level 0 is the desktop window. Level 1 is all top level windows, children of the desktop window. Level two is all children of level 1 windows, etc.

Task-Name

This is the Windows task name. It is the first eight characters of the module name that is obtained from the .DEF file. All task names can be displayed using the Soft-ICE/W **TASK** command.

Module-Name

This is the Windows module name. Windows obtains it from the module's .DEF file. All module names can be displayed using the Soft-ICE/W **MOD** command.

VM-ID

The VM ID is a decimal number that Windows assigns to each virtual machine. Windows uses a VM ID of one for the System VM.

VM Handle

The VM Handle is a 32 bit value that Windows uses to identify each virtual machine. In reality it is the 32 bit address of a structure that contains information about each virtual machine.

VM Status Bits

This is a 32 bit field that contains the status bits for the Virtual Machine. The status bits are defined in the VMM.INC file supplied with the Windows DDK.

VM High Linear Address

Each virtual machine has a high linear address, above 80000000H (2 Gig), that is used to access the memory in a virtual machine. When a VM is the current VM it is mapped both at linear address zero and its high linear address.

Client Register Structure

The Client Register Structure is maintained by the VMM and contains the values of the CPU registers for a particular virtual machine. It can have both a protected mode and a V86 mode section. The exact definition of the Client Register Structure can be found in the VMM.INC file in the DDK.

Selector

This is any valid protected mode selector obtained from the LDT. Selectors can be obtained by using either the **LDT** or the **HEAP** command.

BPR VERBS

R	Break on read access only
W	Break on write access only
RW	Break on any access
T	Backtrace code execution
TW	Backtrace writes to data

File Name

File Name is the full path and file name of a Windows DLL or application.

SYM File Support

Soft-ICE/W can now load .SYM files supplied with the Microsoft Windows SDK. These include KERNEL.SYM, USER.SYM, GDI.SYM and MMSYSTEM.SYM. To pre-load one of these you must use the '**LOAD**' statement in WINICE.DAT with the name of the DLL. For example:

LOAD = C:\WINDOWS\KRNL386.EXE

LOAD = C:\WINDOWS\MMSYSTEM.DLL

If you are using the debugging version of WIN386.EXE, you may also want to load symbols from the WIN386.SYM file. For example:

LOAD = C:\WINDOWS\SYSTEM\WIN386.EXE

WIN386.SYM appears to have symbols that have been exported from the following VXD's:

DEBUG, DOSMGR, DOSNET, EBIOS, AEBIOS, HPEBIOS, VDDCGA, VDDHERC, VDDEGA, VDDVGA30, VDDVGA, VDD8514, VDDCT441, PAGESWAP, PAGEFILE, SHELL, VCD, VDMAD, BLOCKDEV, VKD, VMCD, VMD, VMPOLL, VNETBIOS, VPICD, VSD, VTD, V86MGR, VMM.

Automatic Symbol Table Switching

Soft-ICE/W 1.1 will now automatically attempt to switch to the symbol table that corresponds with the current instruction pointer (cs:eip). This is usually very convenient, however it may be annoying if you have WIN386.SYM loaded. When you pop up Soft-ICE/W you are usually executing in one of the VxD's covered by WIN386.SYM and therefore the current table will be switched to WIN386.SYM instead of the application you are debugging. In cases like this you can turn off the 'auto table switching' feature by entering:

TABLE AUTOOFF

And you can re-enable the feature with:

TABLE AUTOON

If you have auto table switching off, you can switch to the table (if any) that corresponds to the current instruction pointer with:

TABLE \$

Debugging Windows Drivers

Soft-ICE/W 1.0 had problems debugging some Windows drivers at source level. This was caused by limitations in the tools and the structure of the sample assembly drivers supplied by Microsoft. The following is a brief guide to debugging Windows drivers at source level.

Most drivers do not have the problems described below. For most drivers, using the Microsoft tools, you compile with the /Zi switch and link with the /CO switch. If you have problems with source synchronizing up with the generated assembly code, see below.

Many of the sample Windows drivers supplied with the DDK are structured in a way that is not fully compatible with the tools with respect to source level debugging. These problems occur because:

1. Multiple code segments are defined in each module
2. Code is in include files

Problem 1 is handled by a new version of MSYM, supplied with this release. The debug records included in the .EXE file when the /CODEVIEW switch is used do not handle multiple code segments in a module, therefore we must extract the debug information from the .MAP file with the MSYM.EXE utility. MSYM creates a .SYM file that is compatible with Soft-ICE/W. If you have a driver with multiple code segments defined in each module, then build with the following process:

1. MASM with /Zi or /Zd
2. LINK with /LI and /MA (DO NOT USE /CO, it will mess things up)
3. MSYM followed by the file name with no extension
(DO NOT USE Microsoft's MAPSYM, it does not handle source records properly)
4. Place the .SYM file created by MSYM.EXE in the same directory as your driver.
5. Place a "**LOAD** = path\driver-name.DRV" statement in WINICE.DAT.
6. Place a **SYM** = XXXX statement in your WINICE.DAT file, where XXXX is the number of K required for source files and the symbol table.

SOURCE CODE IN INCLUDE FILES

Code in include files is not handled by the linker properly when creating line number records in .MAP files. This problem is present in the sample display drivers that come with the DDK. The worst offender seems to be BITBLT.ASM. If you wish to debug a driver that has code in the include files, there are three alternatives:

1. Eliminate include files by placing the source code from the include file directly in the main module.
2. Eliminate include files by making them separate modules.
3. Fix the .MAP file after linking, by inserting the include file names records into the .MAP file manually with your text editor.

Method 3 is clumsy, and must be done after each link.

The problem is that the line number sections in the .MAP file contain the line number information for the include files, but not the include file names. You must look through the line number sections finding places where a line number is less than the one that preceded it. At this place you must

insert a line that contains the include file name. See file EXAMPLE.MAP for an example of this.

FIXED LOADONCALL SEGMENTS

Some of the segments in the display drivers have the FIXED LOADONCALL attribute. Apparently, these segments may be loaded only in certain Windows modes of operation (standard or enhanced). If you have a segment with this attribute, the symbol table will retain its ordinal number rather than being updated with the actual selector value. To remedy this you must enter SYMLOC R after the segment has been loaded. You can not set break points in these segments until they have been loaded.

Debugging VXD's

Soft-ICE/W 1.0 had problems debugging some VXD's at source level. This was caused by limitations in the tools and the structure of the sample assembly VXD's supplied by Microsoft. The following is a brief guide to debugging Windows VXD's at source level.

Most VXD's do not have the problems described below. For most VXD's, using the Microsoft tools, you compile with the /Zi switch and link with the /CO switch. If you have problems with source synchronizing up with the generated assembly code, read on.

To properly debug a VxD that is being built with the CodeView switch (/CO) on the link line, you may have to alter your source files so that PCODE segments are always placed before ICODE or RCODE in the source files. This is not the case with several sample VxD's supplied with the DDK.

If this re-organization is not reasonable in your situation, you can use the MAPSYM utility to extract the symbol and line number information from the .MAP file. To do this, do the following:

1. Assemble with the /Zi or /Zd switch.
2. Link with /MA /LI (DO NOT USE /CO)
3. Run MSYM.EXE by entering MSYM program-name (with NO extension)
4. Place the .SYM file created by MSYM.EXE in the same directory as your VxD.
5. Place a "**LOAD** = path\vxd-name.386" statement in the WINICE.DAT file.
6. Place a **SYM** = XXXX statement in your WINICE.DAT file, where XXXX is the number of K required for source files and the symbol table.

Debugging Windows Apps and DLL's

In Soft-ICE/W version 1.0 you could not set a break point in a Windows application or DLL until it was actually loaded. Version 1.1 allows break points to be set on symbols or source lines before the program has been loaded. This is especially useful with DLL's that have their symbols loaded with the WINICE.DAT **LOAD** keyword.

In version 1.0 when an application or DLL terminated, Soft-ICE/W deleted all break points and all symbols belonging to the module. Version 1.1 retains all **BPX** and **BPM** style break points and all symbols associated with the module.

PCODE

Segments defined with the **VxD_CODE_SEG** or the **VxD_LOCKED_CODE_SEG** macros.

ICODE

Segments defined with the **VxD_ICODE_SEG** macro.

RCODE

Segments defined with the **VxD_REAL_INIT_SEG** macro.

GRAPHICS

Checking this box in the VDD trapping section will cause Soft-ICE/W to trap all I/O done by the VDD during the switch from text to graphics mode.

TEXT

Checking this box in the VDD trapping section will cause Soft-ICE/W to trap all I/O done by the VDD during the switch from graphics to text mode.

SYMLOC

The SYMLOC command now allows you to change the selector values back to ordinals and to also reset all selectors by looking up their values from ordinals.

To convert all selectors in a symbol table back to ordinals enter:

SYMLOC O

To update all selectors enter:

SYMLOC R

SYMLOC R is most useful if you have a program with some selectors with the FIXED LOADONCALL attributes. These selectors are not created along with the other selectors, and do not get updated in our symbol table. When the selector gets loaded, you can pop-up Soft-ICE/W and do a **SYMLOC R** to turn those ordinal values into their respective selectors.

Miscellaneous changes

LOADX keyword in WINICE.DAT skips source file loading

NOLEDS keyword in WINICE.DAT in case of keyboard lockups

LHEAP command now accepts module name as a parameter

SYMLOC command can switch between ordinals and selectors

.SRC file specifies what source files should be loaded during symbol loading

LOADX

The **LOADX** keyword can be used in WINICE.DAT in place of the **LOAD** keyword. If **LOADX** is specified, Soft-ICE/W will **NOT** load source code for the specified file. This can be useful if there is not enough symbol memory to hold the source code, or if source code is not available for the application.

NOLEDS

Soft-ICE/W has timing problems with some keyboards. If your keyboard is locking up when Soft-ICE/W is popped up, insert the **NOLEDS** keyword in the WINICE.DAT file. The keyword must be placed on a separate line. When the **NOLEDS** keyword is specified, Soft-ICE/W will not send LED commands to the keyboard for the CAPS LOCK and NUM LOCK leds.

LHEAP

A module name can now be used as a parameter to the **LHEAP** command. When a module name is entered, Soft-ICE/W will search for a local heap in the automatic data segment for that module. For example:

LHEAP USER

will display USER.EXE's local heap

Readme Information

This section contains important information that is not in the user manual. Most of this information concerns trouble shooting problems. Also see the troubleshooting section of the Soft-ICE/W manual.

[Installing Soft-ICE/W icons](#)

[Running out of memory](#)

[Loading multiple symbol tables with the LOAD directive .](#)

[Popping up from a serial terminal](#)

[Using Soft-ICE/W on Novell NetWare networks](#)

[Using the debugging version of WIN386.EXE](#)

[VM mode or V86 mode](#)

[QEMM 6.0 users](#)

[386MAX Users](#)

Installing Soft-ICE/W Icons

A program called ICONS.EXE on the release disk will automatically install the icons for the Soft-ICE/W utility programs on a Soft-ICE/W menu. Copy all of the files from the root directory of the distribution diskette(s) to a sub-directory on your hard drive, then run ICONS.EXE from Windows by selecting FILE followed by RUN from the program manager. When RUN prompts for the file name, enter hard-disk-path\ICONS.EXE.

Running out of memory

The default WINICE.DAT reserves 256K of symbol memory with the SYM directive. If you do not need that much memory for symbols and source you can reduce this.

Loading multiple symbol tables with **LOAD** directive

This item concerns loading multiple files from the **LOAD** directive in WINICE.DAT or from the WINICE /load switch. If an error is encountered when loading one of the files, no further symbol table loading is done. If there is a loading problem with one of the files, Soft-ICE/W will prompt you for a key. You can press Ctrl C at this point to exit from Soft-ICE/W back to DOS or press any key to continue. If you continue then all further **LOAD** statements or /load switches will be ignored.

Popping up from a serial terminal

If you are using Soft-ICE/W from a serial terminal, you should place the **COM** keyword in WINICE.DAT. This will allow popping up from the serial terminal. The syntax of the **COM** keyword is:

COMn = ON

Where n is 1, 2, 3 or 4 depending on which COM port will be used. You can alternatively place the /COMn on the Soft-ICE/W command line.

Soft-ICE/W on Novell Netware Networks

When Soft-ICE/W is popped up, all system activity including interrupts is halted. If Soft-ICE/W is popped up for more than 15 minutes the server may disconnect your workstation from the network. Under most conditions this timeout is long enough, but if it becomes a problem you can pop out of Soft-ICE/W and back in every 10 minutes or so.

Under Netware 3.X, you can configure the server to increase its timeout with one or more SET statements on the server's console command line (these are usually put in AUTOEXEC.NCF. The syntax of the SET statement is:

SET name = value

value is in seconds or in packets depending on the name.

There are 3 names that are appropriate:

"NUMBER OF WATCHDOG PACKETS"

limits: 5 to 100

default: 10

"DELAY BETWEEN WATCHDOG PACKETS"

limits: 9.9 seconds to 10 minutes 26.2 seconds

default: 59.3 seconds

"DELAY BEFORE FIRST WATCHDOG PACKET"

limits: 15.7 seconds to 20 minutes 52.3 seconds

default: 4 minutes 56.6 seconds

Example:

SET NUMBER OF WATCHDOG PACKETS = 100

If you enter this command on the server's command line, the total timeout time to be increased from approximately 15 minutes to approximately 2 hours.

Debugging version of WIN386.EXE

If you are developing VxDs and using the debugging version of WIN386.EXE you must use the /K switch on the WINICE command line or the **KBD=TRUE** directive in WINICE.DAT. See APPENDIX B "Using Soft-ICE/W with the Windows Debugging Kernel" in the reference guide for more information.

VM mode or V86 mode

When in a DOS Virtual machine, Soft-ICE/W will display 'V86' in the line above the code window. The reference guide states that 'VM' is displayed.

QEMM 6.0 Users

You cannot use 'stealth' mode in QEMM 6.0 when using Soft-ICE/W. They are being too stealthy, we can't find the video pointers.

.SRC

Creating a .SRC file allows you to specify which source files should be loaded during symbol loading. The default is to load all source files. When Soft-ICE/W is loading symbols, it will check for a file ending with the .SRC extension, that has the same name as the executable file being loaded. If one is found, only source files contained in the .SRC file will be loaded. The .SRC file is an ascii text file containing one or more filenames, one filename per line. Only the file name can be specified, not the drive and directory.

386MAX Users

You cannot use the VGASWAP option in 386MAX. The VGASWAP option relocates the video rom to a different address, usually 0B000H. When this option is used, Soft-ICE/W cannot locate the fonts it needs for screen display. To determine if you are using VGASWAP open the 386MAX.PRO file and look for the VGASWAP keyword.

Soft-ICE/W Diskette

Soft-ICE/W is shipped on either 2 5 1/4" or 1 3 1/2" diskette(s). The distribution diskette(s) contain the following files:

WINICE.EXE	The Soft-ICE/W executable file.
README.1ST	An ascii text file that contains the initial instructions for installing Soft-ICE/W 1.1.
README.HLP	A WINHELP readable file that contains documentation on all changes and new features in Soft-ICE/W 1.1.
A2V.EXE	A DOS program that converts ascii vid files to the .VID format used by Soft-ICE/W.
V2A.EXE	A DOS program that converts .VID files created by Soft-ICE/W to an ascii format.
ET4000.VID	A .VID file for video boards containing the TSENG LABS ET4000 chip set, running WINDOWS 3.1 in standard VGA mode (640x480).
EXAMPLE.MAP	An example of an edited map file that can be used for debugging video drivers.
MSYM.EXE	A DOS program that converts .MAP files into .SYM files for debugging.
WINICE.DAT	The Soft-ICE/W initialization file.
SERIAL.EXE	A DOS program that runs on a remote PC and is used when debugging with a serial terminal.
UPTIME.EXE	A DOS program that sets the time to that of the real time clock.
WLOG.EXE	A DOS/WINDOWS program that dumps the Soft-ICE/W command history buffer to a text file.
VIDMODE.EXE	Configures Soft-ICE/W for CGA, EGA and super VGA video adapters.
VIDMODE.HLP	A WINHELP file that contains on-line help for VIDMODE.
WLDR.EXE	A DOS/WINDOWS program and symbol loader.
WLDR.HLP	A WINHELP file that contains on-line help for WLDR.
ICONS.EXE	A WINDOWS program that creates program manager icons for the Soft-ICE/W utility programs.
COMMDLG3.0	This directory contains the common dialog DLL COMMDLG.DLL for Windows version 3.0.