softice reference V1.10 (fith/March/2001)

(doc version 0.1)
not a cracking tute...

Prelude:

This piece of text was written to chunk together the information a newbie/intermediate cracker may need in order to use softice/reverse engineer... it won't teach you how to crack, its purpose is just to provide a means of reference to commonly used breakpoints, commands, keys and so on.... i'm sure this is not all original text, and would like to point out that all this text is thrown together from my own knowledge and texts i have read/owned...
I know there's texts that tell you all this, but unfortunately i have never seen it all in one file... if there is, sorry for boring you [the reader] with this...
you may also want to check out the usingsoftice.pdf for a better text on using softice (as this is only a brief and quick reference)

==hope it's useful, if not, simply rm iceref.txt==

updates: whenever i can... for requests/suggestions, contact me.
ATM to do: add API description.
Revise text

-you can click the page numbers to go to that page if viewing in Word-

get it running right (winice.dat)

first things first, edit the winice.dat file.
open it up in a text editor and uncomment (remove the ';') the following lines

```
;exp=c:\windows_directory\kernel32.dll
;exp=c:\windows_directory\user32.dll
```

and there you go.. now the 32bit calls can be caught aswell
you can download entire winice.dat files aswell.. premade for you..

in here you will also find the definition of the default keys used in sice...
they are

```
F1  - Help
F2  - Toggle Register Window
F3  - See the dissasembled code mixed with source code (or SRC command)
F5  - Drop back to Windows
F8  - Trace To Next Instruction (will dive into all CALLS)
F10 - Step To Next Instruction (executes and steps over CALLS)
F12 - Run up until the next RETurn instruction
```

Of course you can redefine these, using the command reference at the end of this file.

what do you see? And how to change it

Ok... so we're set.. now we press CTRL+D and what do we see?

first of all, not enough... (possibly)

type these to

```
WR -- show or hide the register window
WD -- show or hide the data window
WC -- show hide the code window
```

If followed by a number will allocate a number of lines to be used for the display

There's a few others, but these are at the end under the complete command reference....

you can use the following keys to scroll the windows into the right position (or use the mouse)

```
<CTRL-UP/DOWN/PGUP/PGDN>  Scrolls the code Window
<SHIFT-UP/DOWN/PGUP/PGDN> Scrolls your INPUT (command) Window
<ALT-UP/DOWN/PGUP/PGDN>   Scrolls the DATA Window
```

Now what does this all mean....?
what you see might make sense to you, but if it doesn’t then read on

the registers should be obvious... eax=<data>, ecx=<data> you know...

flags

then we get the flags... 8 letters on the screen O D I S Z A P C

these mean

O - overflow flag
D - direction flag
I - interrupt carry flag
S - sign flag
Z - zero flag
A - auxilary flag
P - parity flag
C - carry flag

you'll mostly only find the zero flag useful. mostly when you see stuff like jz, jnz etc...

code window

after that the code window... this shows you the snippet of code you are on ATM... this is as follows:

segment:offset opcode asm commands

if you cannot see this, type 'code on' to show it all

breakpoints

ahhh yes... here are the win API calls.... listed by type
(BTW the a at the end of the name denotes a 32bit function call)
set the breakpoints with **bpx**, clear them with **bpc** and disabl them with **bpd**

Reading & Writing Files

These are generic calls to read/write to a file, usually binary in nature:

- ReadFile
- WriteFile

more on locating file accesses:

- SetFilePointer
- GetSystemDirectory
- GetSystemDirectoryA

These are the most common calls to read/write from/to a *.ini file
or a file of similar format.
for 16-bit win apps:

- GetPrivateProfileString
- GetPrivateProfileInt
- WritePrivateProfileString
- WritePrivateProfileInt

for 32-bit win apps:

- GetPrivateProfileStringA
- GetPrivateProfileIntA
- WritePrivateProfileStringA
- WritePrivateProfileIntA

Registry

Create or delete a new key in the registry:

- RegCreateKey
- RegDeleteKey
- RegCreateKeyA
- RegDeleteKeyA

Read a value from the currently open registry key:

- RegQueryValue
- RegQueryValueA
- Open or close a registry key:
- RegCloseKey
- RegOpenKey
- RegCloseKeyA
- RegOpenKeyA

Dialog Boxes

Get text or integer from a dialog box edit:

- GetWindowText
- GetDlgItemText
- GetWindowTextA
- GetDlgItemTextA
- GetDlgItemInt

Open a message box:

- MessageBox
- MessageBoxA

```
      MessageBoxExA
      MessageBeep
```

Time & Date

```
These get the time and date
      GetSystemTime
      GetLocalTime
      SystemTimeToFileTime

Generating a Window
      createwindow
      createwindowexa
      showwindow bitblt (a type of memory move, similar to hmemcpy)
```

CD-ROM Calls

```
      GetDriveType      (if eax=5 then it is a cdrom check)
      GetDriveTypeA

GetDriveType Return Function codes:
   Value               Meaning
     0                 Drive Cannot Be determined
     1                 Root Dir Does not exist
     2                 DriveRemoveable
     3                 A Fixed Disk (HardDrive)
     4                 Remote Drive(Network)
     5                 Cd-Rom Drive
     6                 RamDisk GetLogicalDrives

      GetLogicalDrivesA
      GetLogicalDriveStrings
      GetLogicalDriveStringsA

You know how to set em.. so i won't bother boring you

*tip*
U can use quickview on the target .exe to see what functions from above are
being imported. U can also do this in wdasm... this narrows down the guess work

to deeper understand all this, read up on how the system dlls (containing the
APIs) work...
```

memory searching

```
the syntax is: s <start> l <finish> '<string>'
start is your memlocation you want to start from and finish is... you know...
and string (include the ') is the string you are looking for...
```

breaking on memory access

```
syntax: BPM <address> R/W

this will break on a read or write to memory at <address>
you can do this for a range of memory with
```

BPR <start address> <end address> R/W

Ok that's that...
that'll be your basic break points and sice description/refrence... i said there would be a list of all the sice commands... and here it is... courtesy of ZeroDay, who wrote the following text i stumbled across...

for a more complete reference see the usingice.pdf

complete sice command reference

SETTING BREAKPOINTS:

BPM Breakpoint on memory access

BPMB Breakpoint on memory access

BPMW Breakpoint on memory access

BPMD Breakpoint on memory access

BPR Breakpoint on memory range

BPIO Breakpoint on I/O port access

BPINT Breakpoint on interrupt

BPX Breakpoint on execution

BMSG Breakpoint on windows message

BSTAT Breakpoint statistics

CSIP Set CS:EIP range qualifier

MANIPULATING BREAKPOINTS:

BPE Edit breakpoint

BPT Use breakpoint as a template

BL List current breakpoints

BC Clear Breakpoint

BD Disable breakpoint

BE Enable breakpoint

BH Breakpoint history

DISPLAY/CHANGE MEMORY:

R Display/change register contents

U Un-Assemblers instructions

D Display memory

DB Display memory

DW Display memory

DD Display memory

DS Display memory

DL Display memory

DT Display memory

E Edit memory

EB Edit memory

EW Edit memory

ED Edit memory

ES Edit memory

EL Edit memory

ET Edit memory

PEEK Read from physical address

POKE Write to physical address

H Help on specified function

? Evaluate expression

VER SoftIce version

WATCH Add watch

FORMAT Change format of data window

DATA Change data window

DISPLAY SYSTEM INFORMATION:

GDT Display global descriptor table

LDT Display local descriptor table

IDT Display interrupt descriptor table

TSS Display task state segment

CPU Display CPU register information

PCI Display PCI device information

MOD Display windows module list

HEAP Display windows global heap

LHEAP Display windows local heap

VXD Display windows VxD map

TASK Display windows task list

VCALL Display VxD calls

WMSG Display windows messages

PAGE Display page table information

PHYS Display all virtual addresses for physical address

STACK Display call stack

XFRAME Display active exception frames

MAPV86 Display v86 memory map

HWND Display window handle information

CLASS Display window class information

VM Display virtual machine information

THREAD Display thread information

ADDR Display/change address contents

MAP32 Display 32bit section map

PROC Display process information

QUERY Display processes virtual address space map

WHAT Identify the type of expression

I/O PORT COMMANDS:

I Input data from i/o port

IB Input data from i/o port

IW Input data from i/o port

ID Input data from i/o port

O Output data to i/o port

OB Output data to i/o port

OW Output data to i/o port

OD Output data to i/o port

FLOW CONTROL COMMANDS:

X Return to host debugger or program

G Go to address

T Single step one instruction

P Step skipping calls, Int, etc

HERE Go to current cursor line

EXIT Force an exit to current dos/windows

program

GENINT Generate an interrupt

HBOOT System boot (total reset)

MODE CONTROL:

I1HERE Direct INT1 to SoftIce

I3HERE Direct INT3 to SoftIce

ZAP Zap embedded INT1 or INT3

FAULTS Enable/disable SoftIce fault trapping

SET Change an internal variable

CUSTOMIZATION COMMANDS:

PAUSE Control display scroll mode

ALTKEY Set key sequence to invoke window

FKEY Display/Set function keys

DEX Display/assign window data expression

CODE Display instruction bytes in code window

COLOR Display/set screen colors

ANSWER Auto-answer and redirect console to modem

DIAL Redirect console to modem

SERIAL Redirect console

TABS Set/Display tab settings

LINES Set/display number of lines on screen

PRN Set printer output port

MACRO Define a named macro command

UTILITY COMMANDS:

A Assemble code

S Search for data

F Fill memory with data

M Move data

C Compare two data blocks

WINDOW COMMANDS:

WC Toggle code window

WD Toggle data window

WF Toggle floating point stack window

WL Toggle locals window

WR Toggle register window

WW Toggle watch window

EC Enable/disable code window

. Locate current instruction

WINDOW CONTROL:

CLS Clear window

RS Restore program screen

ALTSCR Change to alternate display

FLASH Restore screen during P and T

SYMBOL/SOURCE COMMANDS:

SYMLOC Relocate symbol base

EXP Display export symbols

SRC Toggle between source,mixed & code

FILE Change/display current source file

SS Search source module for string

TYPES List all types, or display type definition

LOCALS Display locals currently in scope

BACK TRACE COMMANDS:

SHOW Display from backtrace buffer

TRACE Enter backtrace simulation mode

XT Step in trace simulation mode

XP Program step in trace simulation mode

XG Go to address in trace simulation mode

XRSET Reset backtrace history buffer

SPECIAL OPERATORS:

. Preceding a decimal number specifies a line number

$ Preceding an address specifies SEGMENT addressing

Preceding an address specifies SELECTOR addressing

@ Preceding an address specifies indirection

LINE EDITOR KEY USAGE:

[PRINT-SCREEN] Dump Screen to printer

[UP ARROW] Recall previous command line

[DOWN ARROW] Recall next command line

[RIGHT ARROW] Move cursor right

[LEFT ARROW] Move cursor left

[BACKSPACE] Back over last character

[HOME] Start of line

[END] End of line

[INS] Toggle insert mode

[DEL] Delete character

[ESC] Cancel current command

SCROLLING KEY USAGE:

[PAGEUP] Display previous page of display history

[PAGEDOWN] Display next page of display history

[ALT-DN ARROW] Scroll data window down one line

[ALT-UP ARROW] Scroll data window up one line

[ALT-PAGEUP] Scroll data window down one page

[ALT-PAGEDOWN] Scroll data window up one page

[CTRL-UP ARROW] Scroll code window down one line

[CTRL-DN ARROW] Scroll code window up one line

[CTRL-PAGEUP] Scroll code window down one page

[CTRL-PAGEDOWN] Scroll code window up one page

Now the big part of the file... the windows API reference (to my knowledge complete.. You'll find all the APIs here. I haven't yet included descriptions of these APIs, but am working on a newer version that will include –brief-descriptions. If you require more info I suggest you go to the microsoft website (http://msdn.microsoft.com/library/psdk/dasdk/odp475h0.htm) where you should find the complete list and descriptions of up-to-date APIs.

Windows API reference

ACCESSIBILITY

```
GetSystemMetrics
SystemParametersInfo
```

AUDIO

```
auxGetDevCaps
auxGetNumDevs
auxGetVolume
auxSetVolume
PlaySound
sndPlaySound
waveOutGetDevCaps
waveOutGetNumDevs
waveOutGetVolume
waveOutSetVolume
```

BITMAPS

```
BitBlt
ExtFloodFill
GetPixel
SetPixel
SetPixelV
StretchBlt
```

BRUSHES

```
CreateHatchBrush
CreateSolidBrush
GetBrushOrgEx
SetBrushOrgEx
```

COMMON CONTROLS

```
InitCommonControlsEx
```

COMMON DIALOG

```
ChooseColor
ChooseFont
CommDlgExtendedError
GetOpenFileName
GetSaveFileName
PrintDlg
```

CURSOR

```
ClipCursor
CreateCursor
DestroyCursor
GetClipCursor
GetCursor
GetCursorPos
LoadCursor
LoadCursorFromFile
SetCursor
SetCursorPos
SetSystemCursor
ShowCursor
```

DEVICES

```
ChangeDisplaySettings
CreateDC
DeleteDC
DeleteObject
EnumDisplaySettings
GetDC
GetStockObject
ReleaseDC
SelectObject
```

DIALOG BOXES

```
MessageBox
MessageBoxEx
MessageBoxIndirect
```

ERRORS

```
Beep
GetLastError
MessageBeep
SetLastError
SetLastErrorEx
```

FILES

CopyFile
CreateDirectory
CreateDirectoryEx
CreateFile
DeleteFile
FindClose
FindFirstFile
FindNextFile
GetDiskFreeSpace
GetDiskFreeSpaceEx
GetDriveType
GetFileAttributes
GetFileInformationByHandle
GetFileSize
GetFileTime
GetFileVersionInfo
GetFileVersionInfoSize
GetFullPathName
GetLogicalDrives
GetLogicalDriveStrings
GetShortPathName
GetTempFileName
MoveFile
ReadFile
RemoveDirectory
SetFileAttributes
SetFilePointer
SetFileTime
VerQueryValue
WriteFile

FILE SYSTEM

GetVolumeInformation
SetVolumeLabel

FILLED SHAPES

Chord
Ellipse
FillRect
FrameRect
InvertRect
Pie
Polygon
PolyPolygon
Rectangle
RoundRect

FONTS & TEXT

```
CreateFont
CreateFontIndirect
EnumFontFamilies
EnumFontFamiliesEx
GetTextAlign
SetTextAlign
TextOut
```

HANDLES

```
CloseHandle
```

HELP

```
WinHelp
```

ICONS

```
DestroyIcon
DrawIcon
DrawIconEx
ExtractIcon
ExtractIconEx
```

INI FILES

```
GetPrivateProfileInt
GetPrivateProfileString
GetProfileInt
GetProfileString
WritePrivateProfileString
WriteProfileString
```

INPUT (GENERAL)

```
SendInput
```

JOYSTICKS

joyGetDevCaps
joyGetNumDevs
joyGetPos

KEYBOARD

GetAsyncKeyState
GetKeyboardState
GetKeyState
keybd_event
SetKeyboardState

LINES & CURVES

AngleArc
Arc
ArcTo
GetArcDirection
LineTo
MoveToEx
PolyBezier
PolyBezierTo
Polyline
PolylineTo
PolyPolyline
SetArcDirection

MATH

MulDiv

MEDIA CONTROL INTERFACE (MCI)

mciGetErrorString
mciSendString

MEMORY

CopyMemory
FillMemory
GlobalAlloc
GlobalFree
GlobalLock
GlobalMemoryStatus
GlobalMemoryStatusEx

GlobalUnlock
MoveMemory
ZeroMemory

MENUS

CreatePopupMenu
DestroyMenu
GetMenu
GetMenuItemCount
GetMenuItemInfo
GetSystemMenu
InsertMenuItem
RemoveMenu
SetMenuItemInfo
TrackPopupMenu
TrackPopupMenuEx

MESSAGES

SendMessage

MOUSE

GetCapture
GetDoubleClickTime
mouse_event
ReleaseCapture
SetCapture
SetDoubleClickTime
SwapMouseButton

NATIONAL LANGUAGE SUPPORT

GetCurrencyFormat
GetDateFormat
GetNumberFormat
GetThreadLocale
GetTimeFormat
SetThreadLocale

OLE

```
CoTaskMemFree
```

PAINTING & DRAWING

```
GetWindowRgn
SetWindowRgn
```

PENS

```
CreatePen
CreatePenIndirect
```

PRINTERS

```
ClosePrinter
EndDoc
EndPage
EnumJobs
EnumPrinters
OpenPrinter
PrinterProperties
StartDoc
StartPage
```

PROCESSES & THREADS

```
GetEnvironmentVariable
SetEnvironmentVariable
```

RECTANGLES

```
CopyRect
EqualRect
InflateRect
IntersectRect
IsRectEmpty
OffsetRect
PtInRect
SetRect
SetRectEmpty
SubtractRect
```

UnionRect

REGIONS

CombineRgn
CreateEllipticRgn
CreateEllipticRgnIndirect
CreatePolygonRgn
CreatePolyPolygonRgn
CreateRectRgn
CreateRectRgnIndirect
CreateRoundRectRgn
EqualRgn
FillRgn
FrameRgn
GetPolyFillMode
GetRgnBox
InvertRgn
OffsetRgn
PtInRegion
RectInRegion
SetPolyFillMode

REGISTRY

RegCloseKey
RegCreateKeyEx
RegDeleteKey
RegDeleteValue
RegEnumKeyEx
RegEnumValue
RegOpenKeyEx
RegQueryValueEx
RegSetValueEx

SHELL

ExitWindowsDialog
PickIconDlg
RestartDialog
SHAddToRecentDocs
SHBrowseForFolder
Shell_NotifyIcon
ShellExecute
ShellExecuteEx
SHEmptyRecycleBin
SHFileOperation
SHFreeNameMappings
SHGetFileInfo
SHGetFolderLocation
SHGetFolderPath

SHGetPathFromIDList
SHGetSpecialFolderLocation
SHGetSpecialFolderPath
SHQueryRecycleBin
SHUpdateRecycleBinIcon

SHUTDOWN

LockWorkStation

STRINGS

CharLower
CharUpper
CompareString
lstrcmp
lstrcmpi
lstrcpy
lstrcpyn
lstrlen

SYNCHRONIZATION

WaitForSingleObject

SYSTEM INFORMATION

GetComputerName
GetSysColor
GetSystemDirectory
GetTempPath
GetUserName
GetVersionEx
GetWindowsDirectory
SetSysColors

TIME

CompareFileTime
FileTimeToLocalFileTime
FileTimeToSystemTime
GetLocalTime
GetSystemTime

GetSystemTimeAsFileTime
GetTickCount
GetTimeZoneInformation
LocalFileTimeToFileTime
SetSystemTime
SystemTimeToFileTime

TIMERS

KillTimer
QueryPerformanceCounter
QueryPerformanceFrequency
SetTimer

TOOL HELP

CreateToolhelp32Snapshot
Process32First
Process32Next

WINDOW CLASSES

GetClassInfo
GetClassInfoEx
GetClassLong
GetClassName
GetWindowLong
RegisterClass
RegisterClassEx
SetClassLong
SetWindowLong
UnregisterClass

WINDOW PROCEDURES

CallWindowProc
DefWindowProc

WINDOW PROPERTIES

EnumPropsEx
GetProp
RemoveProp
SetProp

WINDOWS

BringWindowToTop
CreateWindowEx
DestroyWindow
EnableWindow
EnumChildWindows
EnumThreadWindows
EnumWindows
FindWindow
FindWindowEx
FlashWindow
GetActiveWindow
GetDesktopWindow
GetFocus
GetForegroundWindow
GetParent
GetTopWindow
GetWindow
GetWindowRect
GetWindowText
GetWindowTextLength
GetWindowThreadProcessId
IsChild
IsIconic
IsWindow
IsWindowEnabled
IsZoomed
MoveWindow
SetActiveWindow
SetFocus
SetForegroundWindow
SetParent
SetWindowPos
SetWindowText
ShowWindow
WindowFromPoint

WINSOCK

```
closesocket
connect
gethostbyaddr
gethostbyname
gethostname
htonl
htons
inet_addr
inet_nota
ioctlsocket NEW
recv
send
socket
WSACleanup
WSAGetLastError
WSAStartup
```

OTHER

```
ExitWindowsEx
Sleep
```

```
==============================
=EOF –c4ffeine@myrealbox.com-=
==============================
```