APPENDIX A

Kernel Debugger Commands

The following tables provide a quick reference to the Windows 2000 Kernel Debugger's console command interface, described in Chapter 1.

TABLE A-1. *Built-in Kernel Debugger Commands*

COMMAND	DESCRIPTION
A [<address>]	Assemble
BA[#] <e|r|w|i><1|2|4> <address>	Address breakpoint
BC[<bp>]	Clear breakpoint(s)
BD[<bp>]	Disable breakpoint(s)
BE[<bp>]	Enable breakpoint(s)
BL[<bp>]	List breakpoint(s)
BP[#] <address>	Set breakpoint
C <range> <address>	Compare memory
D[type][<range>]	Dump memory
E[type] <address> [<list>]	Enter
F <range> <list>	Fill memory
G [=<address> [<address>...]]	Go to address
I<type> <port>	Read from I/O port
J<expression> ['']cmd1['];['']cmd2[']	Conditional execution
K[B] <count>	Stack trace
KB = <base> <stack> <ip>	Stack trace from specific state

(continued)

TABLE A-1. *(continued)*

COMMAND	DESCRIPTION
L{+|-}[lost*]	Control source options
LN <expression>	List nearest symbols
LS[.] [<first>][,<count>]	List source file lines
LSA <address>[,<first>][,<count>]	List source file lines at address
LSC	Show current source file and line
LSF[-] <file>	Load or unload a source file for browsing
M <range> <address>	Move memory
N [<radix>]	Set / show number radix
P[R] [=<address>] [<value>]	Program step
Q	Quit debugger
#R	Multiprocessor register dump
R[F][L][M <expression>] [[<register> [= <expression>]]]	Get/set register/flag value
Rm[?] [<expression>]	Control prompt register output mask
S <range> <list>	Search memory
SS <n | a | w>	Set symbol suffix
SX [e|d [<event>|*|<expression>]]	Exception
T[R] [=<address>] [<expression>]	Trace
U [<range>]	Unassemble
O<type> <port> <expression>	Write to I/O port
X [<*|module>!]<*|symbol>	Examine symbols
.cache [size]	Set virtual memory cache size
.logopen [<file>]	Open new log file
.logappend [<file>]	Append to log file
.logclose	Close log file
.reboot	Reboot target machine
.reload	Reload symbols
~<processor>	Change current processor
? <expression>	Display expression
#<string> [address]	Search for a string in the disassembly
$< <filename>	Take input from a command file

TABLE A-2. *Command Argument Types Used in Table 1-1*

ARGUMENT	DESCRIPTION
<address>	#<16-bit protect-mode [seg:]address>
	&<V86-mode [seg:]address>
<event>	ct, et, ld, av, cc
<expression>	operators: + - * / not by wo dw poi mod(%) and(&) xor(^) or(|) hi low
	operands: number in current radix, public symbol, <register>
<flag>	iopl, of, df, if, tf, sf, zf, af, pf, cf
<list>	<byte> [<byte> ...]
<pattern>	[(nt | <dll-name>)!]<var-name> (<var-name> can include ? and *)
<radix>	8, 10, 16
<range>	<address> <address>
	<address> L <count>
<register>	[e]ax, [e]bx, [e]cx, [e]dx, [e]si, [e]di, [e]bp, [e]sp, [e]ip, [e]fl
	al, ah, bl, bh, cl, ch, dl, dh, cs, ds, es, fs, gs, ss
	cr0, cr2, cr3, cr4, dr0, dr1, dr2, dr3, dr6, dr7
	gdtr, gdtl, idtr, idtl, tr, ldtr
<type>	b (BYTE)
	w (WORD)
	d[s] (DWORD [with symbols])
	q (QWORD)
	f (FLOAT)
	D (DOUBLE)
	a (ASCII)
	c (DWORD and CHAR)
	u (Unicode)
	s|S (ASCII/Unicode string)
	l (list)

TABLE A-3. *Bang Commands Exported by* `kdextx86.dll`

COMMAND	DESCRIPTION		
!acl <Address> [flags]	Display the ACL		
!apic [base]	Dump local APIC		
!arbiter [flags]	Display all arbiters and arbitrated ranges		
	flags:	1	I/O arbiters
		2	Memory arbiters
		4	IRQ arbiters
		8	DMA arbiters
		10	Bus number arbiters
!arblist <address> [flags]	Dump set of resources being arbitrated		
	flags:	1	Include Interface and Slot info per device
!bugdump	Display bug check dump data		
!bushnd	Dump HAL "BUS HANDLER" list		
!bushnd <address>	Dump HAL "BUS HANDLER" structure of handler <address>		
!ca <address> [flags]	Dump control area of a section		
!callback <address> [num]	Dump callback frames for specified thread		
!calldata <table name>	Dump call data hash table		
!cbreg <BaseAddr> | %%<PhyAddr>	Dump CardBus registers		
!cmreslist <CM Resource List>	Dump CM resource list		
!cxr	Dump context record at specified address		
!db <physical address>	Display physical memory BYTEs		
!dblink <address> [count] [bias]	Dump a list via its blinks		
!dcs <Bus>.<Dev>.<Fn>	Dump PCI ConfigSpace of device		
!dd <physical address>	Display physical memory DWORDs		
!defwrites	Dump deferred write queue and triages cached write throttles		
!devext <address> <type>	Dump device extension at <address> of type <type> <type> PCI, PCMCIA, USBD, OpenHCI, USBHUB, UHCD, HID		
!devnode <node> [flags] [service]	Dump device node		
	node:	0	List main tree
		1	List pending removals
		2	List pending ejects
		addr	List specified node
	flags:	1	Dump children
		2	Dump CM Resource List
		4	Dump I/O Resource List
		8	Dump translated CM Resource List
		10	Dump only nodes that aren't started
		20	Dump only nodes that have problems
	service:	If present, only nodes driven by this service dumped	

TABLE A-3. *Bang Commands Exported by* `kdextx86.dll`

COMMAND	DESCRIPTION
!devobj <device>	Dump device object and IRP queue
	<device> Device object address or name
!devstack <device>	Dump device stack associated with device object
!dflink <address> [count] [bias]	Dump a list via its flinks
bias	Mask of bits to ignore in each pointer
!drivers	Display information about all loaded system modules
!drvobj <driver> [flags]	Dump driver object and related information
	<driver> Driver object address or name
	flags: 1 Dump device object list
	2 Dump driver entry points
!eb <physical address> <BYTE list>	Enter BYTE values to physical memory
!ed <physical address>	Enter DWORD values to physical memory <DWORD list>
!errlog	Dump the error log contents
!exca <BasePort>.<SktNum>	Dump ExCA registers
!exqueue [flags]	Dump the ExWorkerQueues
	flags: 1/2/4 Same as !thread / !process
	10 Only critical work queue
	20 Only delayed work queue
	40 Only hypercritical work queue
!exr <address>	Dump exception record at specified address
!filecache	Dump information about the file system cache
!filelock <address>	Dump file lock structure
!filetime	Dump 64-bit FILETIME as a human-readable time
!fpsearch <address>	Find a freed special pool allocation
!frag [flags]	Display kernel mode pool fragmentation
	flags: 1 List all fragment information
	2 List allocation information
	3 Both
!gentable <address>	Dump the given rtl_generic_table
!handle <address> <flags> <process> <TypeName>	Dump handle for a process
	flags: 2 Dump nonpaged object
!heap <address> [flags]	Dump heap for a process
	address: Desired heap to dump or 0 for all
	flags: -v Verbose
	-f Free List entries
	-a All entries

(continued)

TABLE A-3. *Bang Commands Exported by* `kdextx86.dll`

COMMAND	DESCRIPTION
	-s Summary
	-x Force a dump even if the data is bad
!help	Display command help
!HidPpd <address> <flags>	Dump preparsed Data of HID device
!ib <port>	Read a BYTE from an I/O port
!id <port>	Read a DWORD from an I/O port
!ioapic [base]	Dump I/O APIC
!ioreslist <IO Resource List>	Dump I/O resource requirements list
!irp <address> <dumplevel>	Dump IRP at specified address
	address == 0 Dump active IRPs (checked only)
	dumplevel:0 Basic stack info
	1 Full field dump
	2 Include tracking information (checked only)
!irpfind [pooltype] [restart addr] [<irpsearch> <address>]	Search pool for active IRPs
	pooltype: 0 Nonpaged pool (default)
	1 Paged pool
	2 Special pool
	restart addr If present, scan will be restarted from here in pool
	<irpsearch> Specifies filter criteria to find a specific IRP:
	userevent Irp.UserEvent == <address>
	device Stack location: DeviceObject == <address>
	fileobject Irp.Tail.Overlay.OriginalFileObject == <address>
	mdlprocess Irp.MdlAddress.Process ==<address>
	thread Irp.Tail.Overlay.Thread == <address>
	arg One of the arguments == <address>
!iw <port>	Read a WORD from an I/O port
!job <address> [<flags>]	Dump JobObject at <address>, processes in job
!locks [-v] <address>	Dump kernel-mode resource locks
!lookaside <address> <options> <depth>	Dump lookaside lists
	options: 1 Reset list counters
	2 Set list depth to <depth>
!lpc	Dump LPC ports and messages
!memusage	Dump the page frame database table
!mps	Dump MPS BIOS structures
!mtrr	Dump MTTR
!npx [base]	Dump NPX save area
!ob <port>	Write a BYTE to an I/O port

TABLE A-3. *Bang Commands Exported by* `kdextx86.dll`

COMMAND	DESCRIPTION
!obja <TypeName>	Dump an object manager object's attributes
!object <-r | Path | address | 0 TypeName>	Dump an object manager object
	-r Force reload of cached object pointers
!od <port>	Write a DWORD to an I/O port
!ow <port>	Write a WORD to an I/O port
!patch	Enable and disable various driver flags
!pci [flag] [bus] [device] [function]	Dump PCI type1 configuration [rawdump:minaddr] [maxaddr]
	flag: 0x01 Verbose
	0x02 From bus 0 to 'bus'
	0x04 Dump raw BYTEs
	0x08 Dump raw DWORDs
	0x10 Do not skip invalid devices
	0x20 Do not skip invalid functions
	0x40 Dump capabilities if found
	0x80 Dump device specific on VendorID:8086
!pciir	Dump the PCI IRQ routing table
!pcitree	Dump the PCI tree structure
!pcr	Dump the Processor Control Region (PCR)
!pfn	Dump the page frame database entry for the physical page
!pic	Dump PIC (8259) information
!pnpevent <address>	Dump specified PNP event, or all events if <address> == 0
!pocaps	Dump system power capabilities
!podev <devobj>	Dump power relevant data in device object
!polist	Dump power IRP serial list
!polist [<devobj>]	Dump power IRP serial list entries for specified devobj
!ponode	Dump power device node stack (devnodes in power order)
!popolicy	Dump system power policy
!poproc <Address>	Dump processor power state.
!pool <address> [detail]	Dump kernel mode heap
	address: 0 Only the process heap (default)
	-1 All heaps in the process
	else Pool entry
	detail: 0 Summary Information
	1 Summary + location/size of regions
	2 Display information only for address

(continued)

TABLE A-3. *Bang Commands Exported by* `kdextx86.dll`

COMMAND	DESCRIPTION		
		3	Summary + blocks in committed regions
		4	Summary + free lists
!poolfind <tag> [pooltype]	Find occurrences of the specified pool <tag>		
	<tag>	Four-character tag, * and ? are wild cards	
	pooltype:	0	Nonpaged pool (default)
		1	Paged pool
		2	Special pool
!poolused [flags [TAG]]	Dump usage by pool tag		
	flags:	1	Verbose
		2	Sort by NonPagedPool Usage
		4	Sort by PagedPool Usage
!poReqList [<devobj>]	Dump PoRequestedPowerIrp created Power IRPs		
!portcls <devobj> [flags]	Dump portcls data for portcls bound devobj		
	flags:	1	Port dump
		2	Filter dump
		4	Pin dump
		8	Device context
		10	Power info
		100	Verbose
		200	Really verbose
!potrigger <address>	Dump POP_ACTION_TRIGGER		
!process [flags] [image name]	Dump process at specified address		
	flags:	1	Don't stop after Cid/Image information
		2	Dump thread wait states
		4	Dump only thread states
		6	Dump thread states and stack
!processfields	Show offsets to all fields in the EPROCESS structure		
!pte	Dump the corresponding PDE and PTE for the entered address		
!ptov <PhysicalPageNumber>	Dump all valid physical/virtual mappings for a page directory		
!qlocks	Dump state of all queued spin locks		
!range <RtlRangeList>	Dump RTL_RANGE_LIST		
!ready	Dump state of all ready system threads		
!reghash	Dump registry hash table		
!regkcb <address>	Dump registry key-control-blocks		
!regpool [s|r]	Dump registry allocated paged pool		
	s	Save list of registry pages to temporary file	
	r	Restore list of registry pages from temporary file	

TABLE A-3. *Bang Commands Exported by* `kdextx86.dll`

COMMAND	DESCRIPTION
!rellist <relation list> [flags]	Dump PNP relation lists
	flags: 1 Not used
	2 Dump CM Resource List
	4 Dump I/O Resource List
	8 Dump translated CM Resource List
!remlock	Dump a remove lock structure
!sd <Address> [flags]	Display SECURITY_DESCRIPTOR
!sel [selector]	Examine selector values
!session <id> [flags] [image name]	Dump sessions
!sid <Address> [flags]	Display the SID structure at the specified address
!socket <address>	Dump PCMCIA socket structure
!srb <address>	Dump SCSI Request Block at specified address
!stacks <detail-level>	Dump summary of current kernel stacks
	detail-level: 0 Display stack summary
	1 Display stacks, no parameters
	2 Display stacks, full parameters
!sysptes	Dump the system PTEs
!thread <address> [flags]	Dump thread at specified address
	flags: 1 Not used
	2 Dump thread wait states
	4 Dump only thread states
	6 Dump thread states and stack
!threadfields	Show offsets to all fields in the ETHREAD structure
!time	Report PerformanceCounterRate and TimerDifference
!timer	Dump timer tree
!token <address> [flags]	Dump token at specified address
!tokenfields	Show offsets to all fields in a token structure
!trap [base]	Dump trap frame
!tss [register]	Dump TSS
!tunnel <address>	Dump a file property tunneling cache
!tz [<address> <flags>]	Dump thermal zones (No arguments: dump all zones)
!tzinfo <address>	Dump thermal zone information
!urb <address> <flags>	Dump an USB Request Block
!usblog <log> [addr] [flags]	Display an USB log
	<log> USBHUB, USBD, UHCD, OpenHCI

(continued)

TABLE A-3. *Bang Commands Exported by* `kdextx86.dll`

COMMAND	DESCRIPTION		
	addr:		Address to begin dumping from in <log>
	flags:	-r	Reset the log to dump from most recent entry
		-s L	Search for tags in comma-delimited list L
		-l N	Set number of lines to display at a time to N
!usbstruc <address> <type>	Display an USB HC descriptor of <type>		
	<type>	OHCIReg, HCCA, OHCIHcdED, OHCIHcdTD, OHCIEndpoint, DevData, UHCDReg	
!vad	Dump VADs		
!version	Version of extension DLL		
!vm	Dump virtual memory values		
!vpd <address>	Dump volume parameter block		
!vtop DirBase address	Dump physical page for virtual address		
!wdmaud <address> <flags>	Dump wdmaud data for structures		
	flags:	1	I/O control history dump given WdmaIoctlHistoryListHead
		2	Pending IRPs given WdmaPendingIrpListHead
		4	Allocated MDLs given WdmaAllocatedMdlListHead
		8	pContext dump given WdmaContextListHead
		100	Verbose
!zombies	Find all zombie processes		

TABLE A-4. *Bang Commands Exported by* `userkdx.dll`

COMMAND	DESCRIPTION
!atom	Dump atoms or atom tables
!dcls [pcls]	Dump window class
!dcss	Dump critical section stack traces
!dcur -aivp [pcur]	Dump cursors
!dde -vr [conv|window|xact]	Dump DDE tracking information
!ddesk -vh <pdesk>	Display objects allocated in desktop
!ddl [pdesk]	Dump desktop log
!ddk <pKbdTbl>	Dump deadkey table
!df [flags] | [-p pid]	Display or set debug flags
!dfa	Dump allocation fail stack trace
!dha address	Dump heap allocations and verify heap
!dhe [pointer|handle] | [-t[o[p]] type [pti/ppi]]	Dump handle entries
!dhk -ag [pti]	Dump hooks
!dhot	Dump registered hotkeys
!dhs -vpty [id|type]	Dump handle table statistics
!di	Display USER input processing globals
!dii <piiex>	Dump extended IME information
!dimc [-hrvus] -[wci] [imc|wnd,etc.]	Dump Input Context
!dimk [pImeHotKeyObj]	Dump IME Hotkeys
!dinp -v [pDeviceInfo]	Dump input diagnostics
!dkl -akv <pkl>	Dump keyboard layout structures
!dll [*]addr [l#] [b#] [o#] [c#] [t[addr]]	Dump linked list (can Ctrl-C)
!dlr <pointer|handle>	Display assignment locks for object
!dm -vris <menu|window>	Dump a menu
!dmon <pMonitor>	Dump MONITOR
!dmq [-ac] [pq]	List messages in queues
!dms <MenuState>	Dump a pMenuState
!dp -vcpt [id]	Display simple process information
!dpa -cvsfrp	Dump pool allocations
!dpi [ppi]	Display PROCESSINFO structure specified
!dpm <ppopupmenu>	Dump a popup menu
!dq -t [pq]	Display Q structure specified
!dsbt <pSBTrack>	Display Scroll Bar Track structure
!dsbwnd <psbwnd>	Dump extra fields of Scrollbar windows
!dsi [-bchmopvw]	Display SERVERINFO structure

(continued)

TABLE A-4. *(continued)*

COMMAND	DESCRIPTION
!dsms -vl [psms]	Display SMS (SendMessage structure) specified
!dso <Structure> [Field] [addr [*n]]	Dump structure field(s)'s offset(s) and value(s)
!dt -gvcp [id]	Display simple thread information
!dtdb [ptdb]	Dump Task Database
!dti [pti]	Display THREADINFO structure
!dtl [-t] [pointer|handle]	Display thread locks
!dtmr [ptmr]	Dump timer structure
!du [pointer|handle]	Generic object dumping routine
!dumphmgr [-s]	Dump object allocation counts (debug version only)
!dup	User preferences DWORDs
!dupm	User preference bit mask
!dvs -s	Dump sections and mapped views
!dw -aefhvsprwoz [hwnd/pwnd]	Display information on windows in system
!dwe [-n] [addr]	Display WinEvent hooks/notifies
!dwpi -p [pwpi | ppi]	Display WOWPROCESSINFO structure specified
!dws [pws]	Dump window stations
!dy [pdi]	Dump DISPLAYINFO
!find baseaddr addr [o#]	Find linked list element
!fno <address>	Find nearest object
!frr <psrcLo> <psrcHi> <prefLo> [prefHi]	Find Range Reference
!help -v [cmd]	Display command help (-v verbose)
!hh	Dump gdwHydraHint
!kbd -au [pq]	Display key state for queue
!sas [-s] <addr> [length]	Stack Analysis Stuff
!test	Test basic debug functions
!uver	Show versions of USERTEXTS and WIN32K.SYS