Microsoft

Windows Embedded Compact 7
Boot Time Performance

Douglas Boling

Boling Consulting Inc.

Windows Embedded Compact 7

About Douglas Boling

- Independent consultant specializing in Windows Mobile and Windows Embedded Compact (Windows CE)
  - On-Site Instruction
  - Consulting and Development
- Author
  - *Programming Embedded Windows CE*
    - *Fourth Edition*

Agenda

- Boot sequence explained
- Instrumenting the boot
- Speeding up the boot sequence

The Boot Sequence Summary

- Machine Startup
- Kernel Boot
- System Startup

Windows Embedded Compact 7

Machine Startup

- BIOS/EFI startup
  - On x86 and some ARM systems
  - For fast boot, disable startup tests
- Boot loader launch
  - If NAND flash
    - initial program load (IPL) code reads boot loaded into RAM
  - If Disk based
    - Boot sector in disk boot partition finds boot loader and reads it into RAM

Kernel Boot

- Bootloader loads operating system Image into RAM
  - Typically NK.BIN
  - Speed depends on the speed of flash and the size of the image
- Initial Kernel startup
  - Typically quite fast
  - Kernel (NK) and FileSys modules loaded and initialized
  - This is the place to configure metering code

System Startup

- Driver loading
  - Drivers load serially
  - User mode drivers load in unique driver manager instances
  - Driver load driven by registry
- Services startup
  - Services load serially
  - Very similar to driver initialization
  - Network access typical

System Startup (2)

- Shell startup
  - Explorer
  - XAML-based “Home screen”
  - Thin client shell
- Application startup
  - If Explorer shell
    - Applications in Startup Folder launched
  - Otherwise, launch driven by registry

Configuring the Boot Sequence

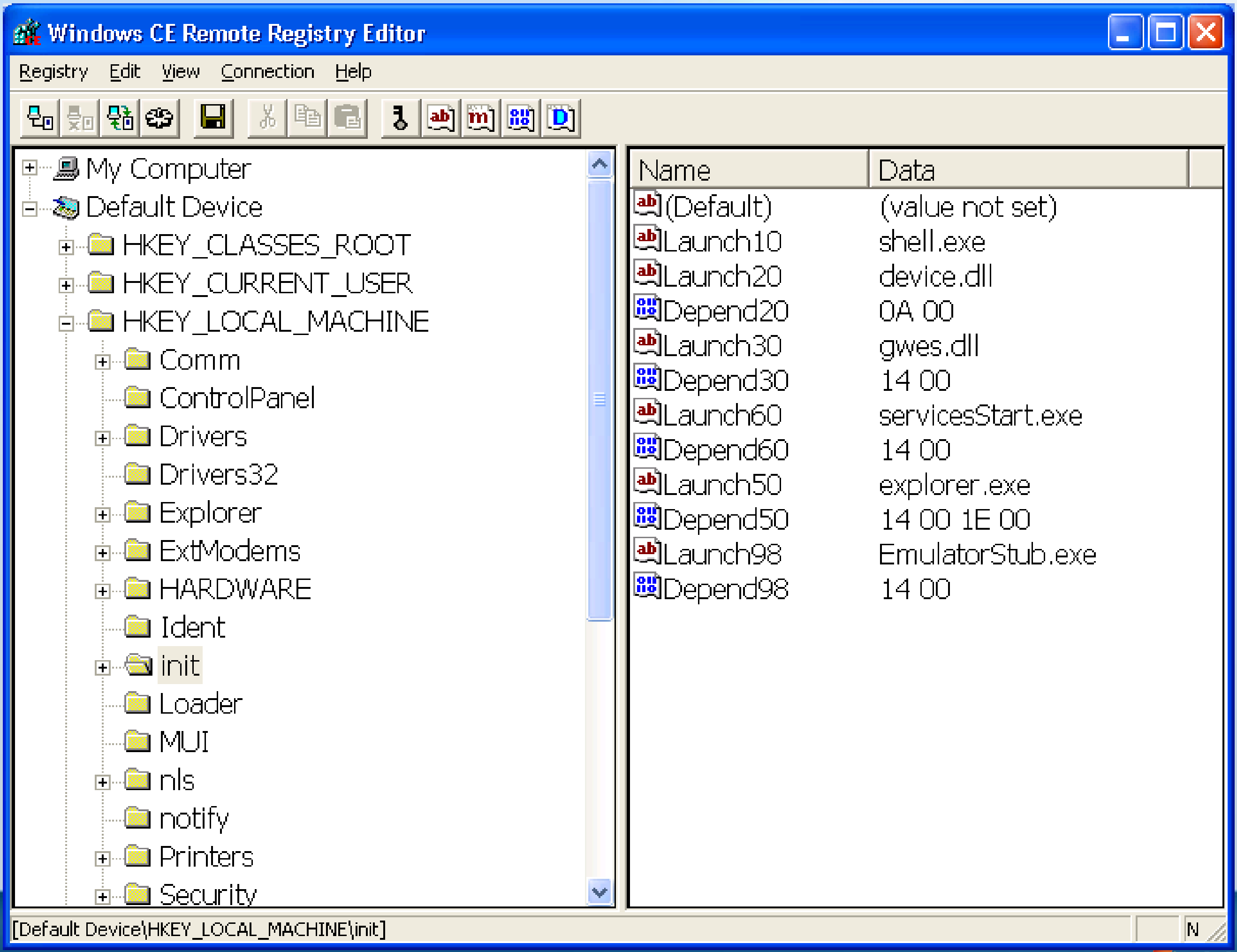
- Machine Startup
  - Configured in BIOS / EFI settings
  - IPL or boot sector code
  - Boot loader code
- Kernel Boot
  - Not really configurable
  - Componentization decisions can help
- System Startup
  - In the registry

Registry Configuration

- System Boot sequence configured in [HKLM\Init]
  - Configures the launch sequence of the operating system
  - Values "Launchxx" define kernel DLLs and Applications to load
  - Values "Dependxx" serialize the load sequence to support dependencies

- Driver load sequence configured in [HKLM\Drivers\BuiltIn]
  - Each subkey represents a driver to load
  - The "Order" value in the subkeys define the load order of the drivers

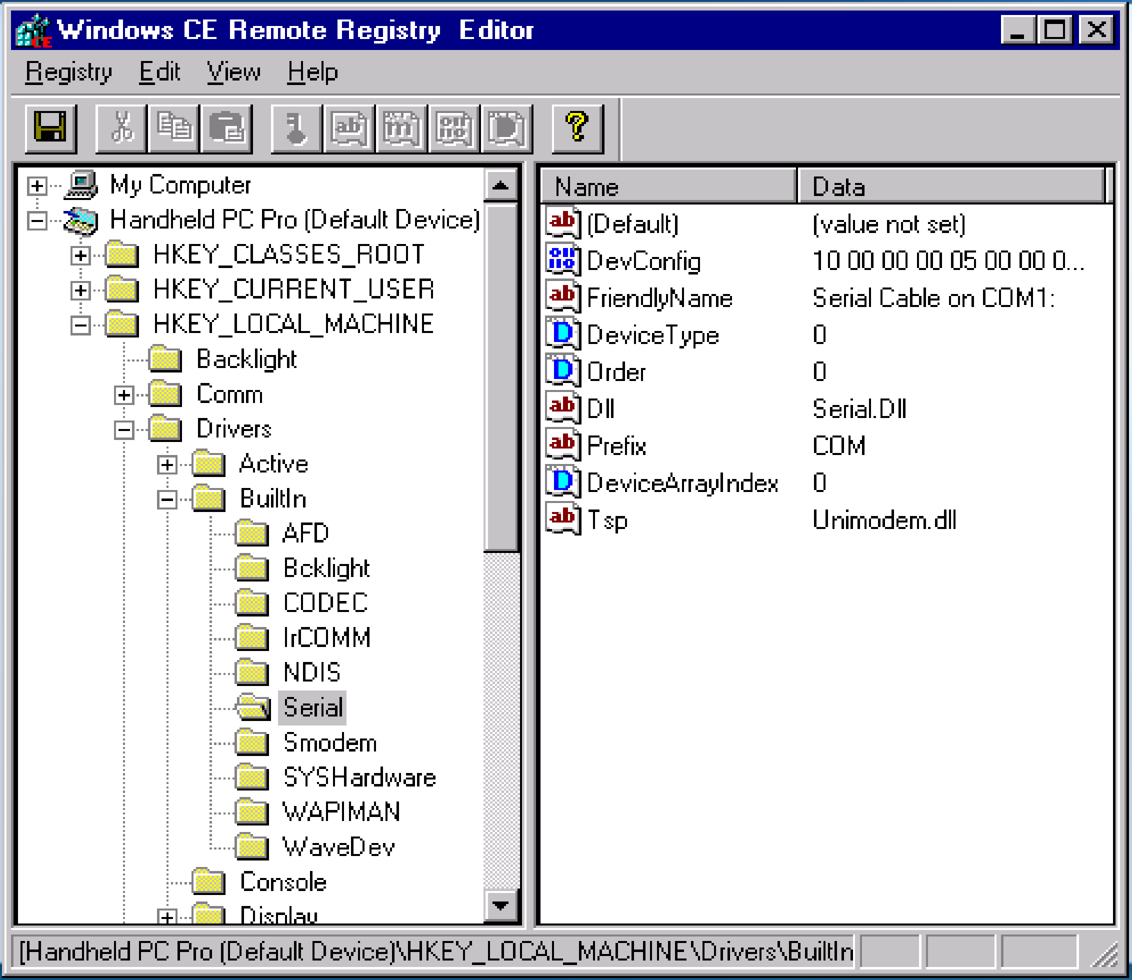
HKEY_LOCAL_MACHINE\Init



Name	Data
(Default)	(value not set)
Launch10	shell.exe
Launch20	device.dll
Depend20	0A 00
Launch30	gwes.dll
Depend30	14 00
Launch60	servicesStart.exe
Depend60	14 00
Launch50	explorer.exe
Depend50	14 00 1E 00
Launch98	EmulatorStub.exe
Depend98	14 00

[Default Device\HKEY_LOCAL_MACHINE\init]

[HKLM]\Drivers\BuiltIn



- My Computer
- Handheld PC Pro (Default Device)
  - HKEY_CLASSES_ROOT
  - HKEY_CURRENT_USER
  - HKEY_LOCAL_MACHINE
    - Backlight
    - Comm
    - Drivers
      - Active
      - BuiltIn
        - AFD
        - Bcklight
        - CODEC
        - IrCOMM
        - NDIS
        - Serial
        - Smodem
        - SYSHardware
        - WAPIMAN
        - WaveDev
      - Console
      - Display

Name	Data
(Default)	(value not set)
DevConfig	10 00 00 00 05 00 00 0...
FriendlyName	Serial Cable on COM1:
DeviceType	0
Order	0
Dll	Serial.Dll
Prefix	COM
DeviceArrayIndex	0
Tsp	Unimodem.dll

[Handheld PC Pro (Default Device)\HKEY_LOCAL_MACHINE\Drivers\BuiltIn

The Key to Boot Optimization

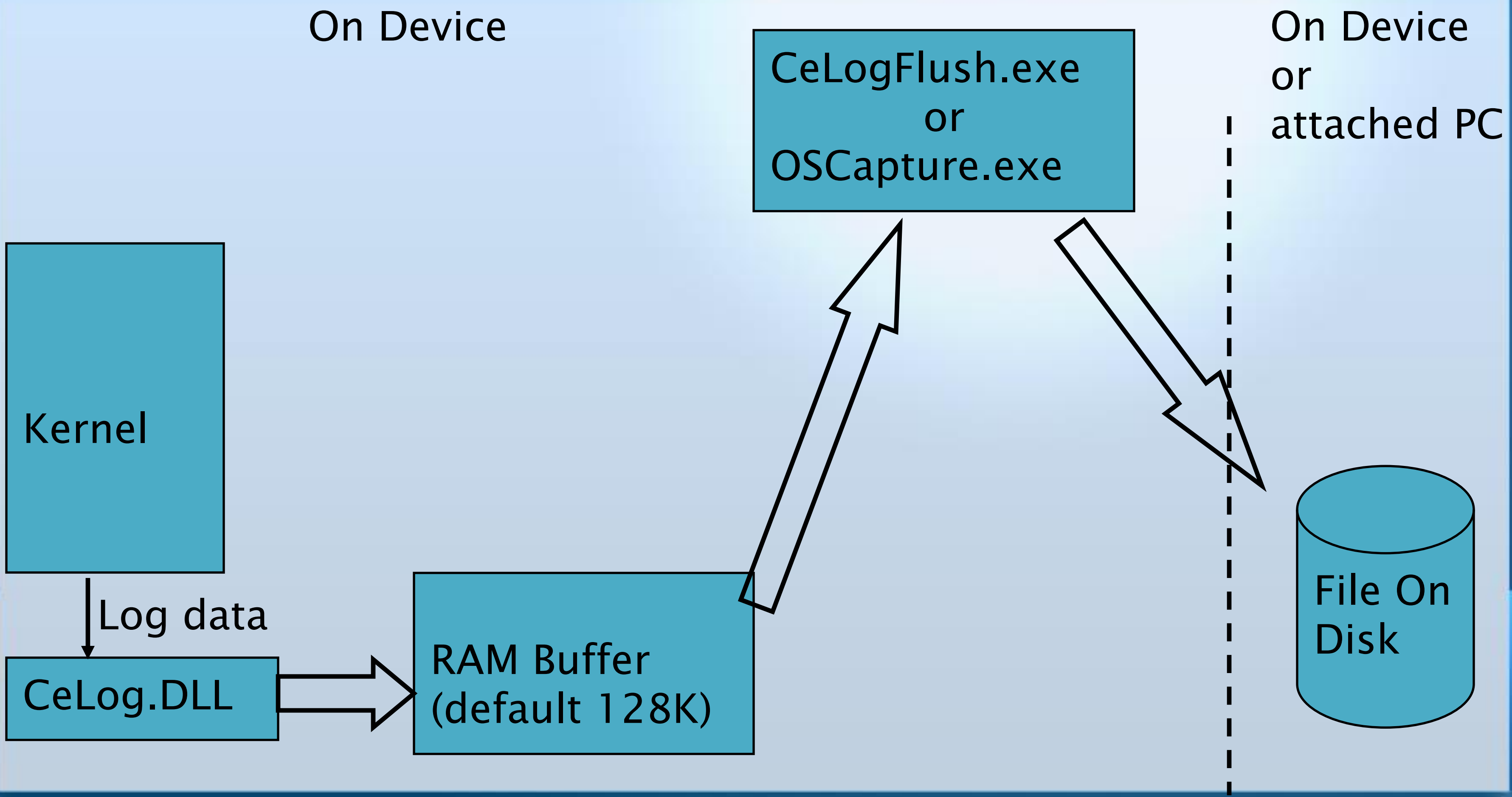
Know what is going on!

Know what is going on!

- The CeLog tool is great for this
  - CELog is a kernel level logging infrastructure
- Will log everything
  - Interrupts
  - TLB misses (on MIPs and SH4 CPUs)
  - Thread switches
  - Memory allocations
  - Kernel sync objects
  - OEM defined events

Windows Embedded Compact 7

CeLog Architecture



CELog Internal Design

- CELog collects information in circular memory buffer
  - 128 KB by default
  - Configurable at load via registry
- Flush utility (CELogFlush) copies data in buffer to file
  - File is in root or release directory
  - *.clg files
  - Source in public\common\sdk\samples\CeLog\flush
- Another utility (OSCapture) copies data to file on device
  - Same file format

CELog Zones

CELZONE_INTERRUPT	0x00000001	*Events related to interrupts.*
CELZONE_RESCHEDULE	0x00000002	*Events related to the scheduler.*
CELZONE_MIGRATE	0x00000004	*Events related to migration of threads between processes.*
CELZONE_TLB	0x00000008	*Events related to translation look-aside buffer (TLB). (MIPS and SH4)*
CELZONE_DEMANDPAGE	0x00000010	*Events related to paging.*
CELZONE_THREAD	0x00000020	*Events related to threads, except for thread switches.*
CELZONE_PROCESS	0x00000040	*Events related to processes.*
CELZONE_PRIORITYINV	0x00000080	*Events related to priority inversion.*
CELZONE_CRITSECT	0x00000100	*Events related to critical sections.*
CELZONE_SYNCH	0x00000200	*Events related to synchronization.*
CELZONE_PROFILER	0x00000400	*Events related to profiling.*
CELZONE_HEAP	0x00000800	*Events related to heaps.*
CELZONE_VIRTMEM	0x00001000	*Events related to virtual memory.*
CELZONE_GWES	0x00002000	*Events related to the Graphics, Windowing, and Event system.*
CELZONE_LOADER	0x00004000	*Events related to the loader.*
CELZONE_MEMTRACKING	0x00008000	*Events related to memory tracking.*
CELZONE_BOOT_TIME	0x00010000	*Events in the boot process*
CELZONE_GDI	0x00020000	*Events related to GDI.*
CELZONE_KCALL	0x00400000	*Events related to KCALLs. Used by profilier*
CELZONE_DEBUG	0x00800000	*Duplicate debug output strings in log.*

Zones For Boot Time Performance – 0x14266

CELZONE_INTERRUPT	0x00000001	Events related to interrupts.
CELZONE_RESCHEDULE	0x00000002	Events related to the scheduler.
CELZONE_MIGRATE	0x00000004	Events related to migration of threads between processes.
CELZONE_TLB	0x00000008	Events related to the translation look-aside buffer (TLB).
CELZONE_DEMANDPAGE	0x00000010	Events related to paging.
CELZONE_THREAD	0x00000020	Events related to threads, except for thread switches.
CELZONE_PROCESS	0x00000040	Events related to processes.
CELZONE_PRIORITYINV	0x00000080	Events related to priority inversion.
CELZONE_CRITSECT	0x00000100	Events related to critical sections.
CELZONE_SYNCH	0x00000200	Events related to synchronization.
CELZONE_PROFILER	0x00000400	Events related to profiling.
CELZONE_HEAP	0x00000800	Events related to heaps.
CELZONE_VIRTMEM	0x00001000	Events related to virtual memory.
CELZONE_GWES	0x00002000	Events related to the Graphics, Windowing, and Event system.
CELZONE_LOADER	0x00004000	Events related to the loader.
CELZONE_MEMTRACKING	0x00008000	Events related to memory tracking.
CELZONE_BOOT_TIME	0x00010000	Events in the boot process
CELZONE_GDI	0x00020000	Events related to GDI.
CELZONE_KCALL	0x00400000	Events related to KCALLs.
CELZONE_DEBUG	0x00800000	Duplicate debug output strings in log.

Using CELog

- CELog needs the CELog.DLL in the image
  - Needs to be there when kernel starts
- Mask unneeded logging zones to reduce data
- Enlarge RAM buffer to eliminate data loss
- Start CeLogFlush on boot
- CeLog source in private directory
  - Private\winceos\coreos\nk\celog

Including CeLog in Image

- Configure image by setting

```
REM Include CeLog files in image
Set IMGCELOGENABLE=1

REM Configure CeLogFlush to launch after FileSys
Set IMGAUTOFLUSH=1
```

- To use OsCapture.EXE instead of CeLogFlush

```
REM Configure OsCapture to launch after FileSys
Set IMGOSCAPTURE=1
```

- Don’t set both IMGAUTOFLUSH and IMGOSCAPTURE

CeLog Configuration at Boot Time

- Registry not available when CeLog.DLL loads at boot
- CeLog buffer size main issue
  - Defaults to 128K, easily too small
- Embedded CE 6: Buffer size can not be configured
  - Should work with FIXUPVAR to dwCeLogLargeBuf however variable wasn't marked "const volatile" and was optimized out
  - Solution: rebuild kernel.dll or CeLog.dll to change
- Compact 7: OAL fields IOCTL_HAL_GET_CELOG_PARAMETERS

IOCTL_HAL_GET_CELOG_PARAMETERS

- IOCTL sent to OAL from CeLog DLL on boot
  - Pointer to OEMCeLogParameters structure passed in Output buffer

```
typedef struct {
    DWORD dwVersion;              // Version of this structure, set to 1
    DWORD MainBufferAddress;      // Virtual address for buffer (0 for no address)
    DWORD MainBufferSize;         // Size of the buffer
    DWORD SyncDataSize;           // Portion of the main buffer to use for
                                  //   thread/process/module info
    BOOL  ClearExistingData;      // Says whether to wipe buffer from a previous boot
                                  //   (only used if MainBufferAddress != 0)
    BOOL  AutoEraseMode;          // Indicates to discard old data to make room for new
    DWORD ZoneCE;                 // CeLog zone settings
} OEMCeLogParameters_V1;
```

- Structure prepopulated.
  - Only update what you need to change

CELogFlush Registry Entries

- CeLogFlush registry entries

```
[HKEY_LOCAL_MACHINE\System\CeLog]
    "FileName“ = <Path & file name of .clg file>
    "Transport"= "Local File" | "RAM" | “CESH"
    "FlushTimeout"= dword:<flush timeout in mS>
    “FileSize"= dword:<Max size of .clg before new file>
    “FileFlags"= dword:<0, 1, 2>
       0 = Close .clg file after some idle time (def.)
       1 = Never close .clg file
       2 = Close .clg file after every flush
    "ThreadPriority“= dword:<flush thread priority>
```

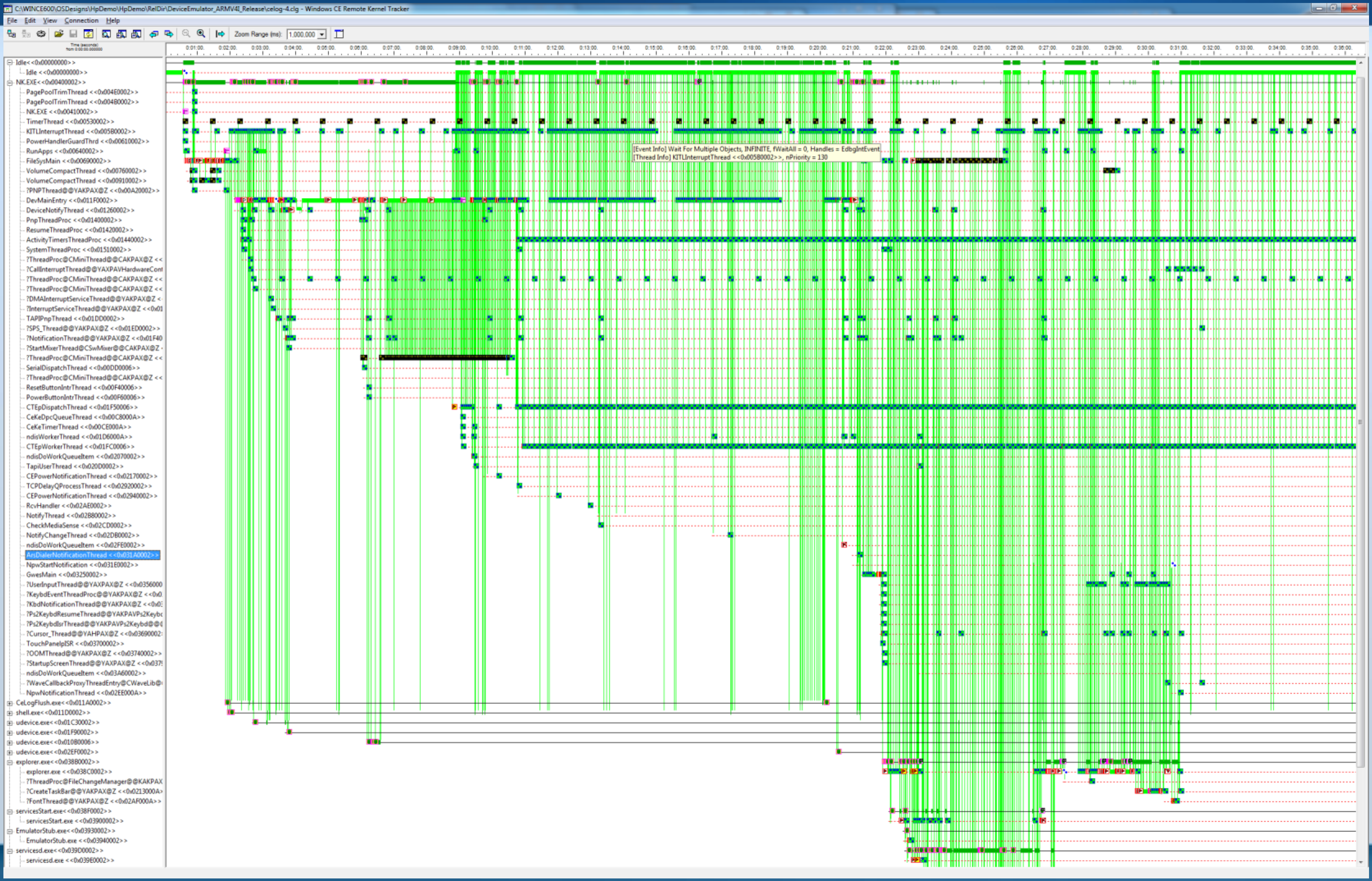
Windows Embedded Compact 7

- Demo

Using CeLog for Boot Analysis

Windows Embedded Compact 7

Kernel Tracker View of Boot

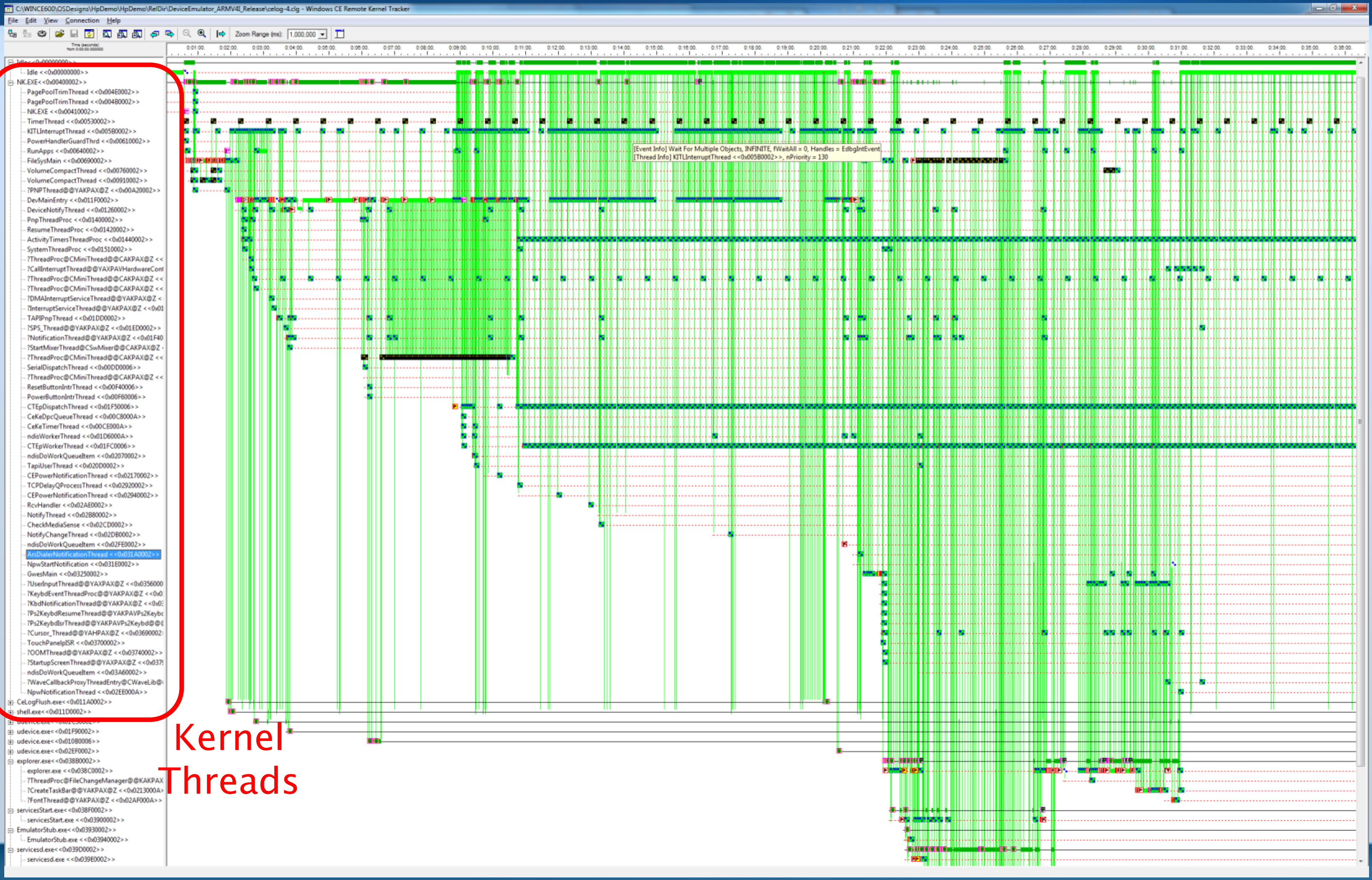

C:\WINCE600\OSDesigns\HpDemo\HpDemo\RelDir\DeviceEmulator_ARMV4I_Release\celog-4.clg - Windows CE Remote Kernel Tracker

```
Idle<<0x00000000>>
  Idle <<0x00000000>>
NK.EXE<<0x00400002>>
  PagePoolTrimThread <<0x004E0002>>
  PagePoolTrimThread <<0x004B0002>>
  NK.EXE <<0x00410002>>
  TimerThread <<0x00530002>>
  KITLInterruptThread <<0x005B0002>>
  PowerHandlerGuardThrd <<0x00610002>>
  RunApps <<0x00640002>>
  FileSysMain <<0x00690002>>
  VolumeCompactThread <<0x00760002>>
  VolumeCompactThread <<0x00910002>>
  ?PNPThread@@YAKPAX@Z <<0x00A20002>>
  DevMainEntry <<0x011F0002>>
  DeviceNotifyThread <<0x01260002>>
  PnpThreadProc <<0x01400002>>
  ResumeThreadProc <<0x01420002>>
  ActivityTimersThreadProc <<0x01440002>>
  SystemThreadProc <<0x01510002>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?CallInterruptThread@@YAXPAVHardwareCont
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?DMAInterruptServiceThread@@YAKPAX@Z <
  ?InterruptServiceThread@@YAKPAX@Z <<0x01
  TAPIPnpThread <<0x01DD0002>>
  ?SPS_Thread@@YAKPAX@Z <<0x01ED0002>>
  ?NotificationThread@@YAKPAX@Z <<0x01F40
  ?StartMixerThread@CSwMixer@@CAKPAX@Z
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  SerialDispatchThread <<0x00DD0006>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ResetButtonIntrThread <<0x00F40006>>
  PowerButtonIntrThread <<0x00F60006>>
  CTEpDispatchThread <<0x01F50006>>
  CeKeDpcQueueThread <<0x00C8000A>>
  CeKeTimerThread <<0x00CE000A>>
  ndisWorkerThread <<0x01D6000A>>
  CTEpWorkerThread <<0x01FC0006>>
  ndisDoWorkQueueItem <<0x02070002>>
  TapiUserThread <<0x020D0002>>
  CEPowerNotificationThread <<0x02170002>>
  TCPDelayQProcessThread <<0x02920002>>
  CEPowerNotificationThread <<0x02940002>>
  RcvHandler <<0x02AE0002>>
  NotifyThread <<0x02B80002>>
  CheckMediaSense <<0x02CD0002>>
  NotifyChangeThread <<0x02DB0002>>
  ndisDoWorkQueueItem <<0x02FE0002>>
  ArsDialerNotificationThread <<0x031A0002>>
  NpwStartNotification <<0x031E0002>>
  GwesMain <<0x03250002>>
  ?UserInputThread@@YAXPAX@Z <<0x0356000
  ?KeybdEventThreadProc@@YAKPAX@Z <<0x0
  ?KbdNotificationThread@@YAKPAX@Z <<0x0
  ?Ps2KeybdResumeThread@@YAKPAVPs2Keybd
  ?Ps2KeybdIsrThread@@YAKPAVPs2Keybd@@
  ?Cursor_Thread@@YAHPAX@Z <<0x03690002
  TouchPanelpISR <<0x03700002>>
  ?OOMThread@@YAKPAX@Z <<0x03740002>>
  ?StartupScreenThread@@YAXPAX@Z <<0x037
  ndisDoWorkQueueItem <<0x03A60002>>
  ?WaveCallbackProxyThreadEntry@CWaveLib@
  NpwNotificationThread <<0x02EE000A>>
CeLogFlush.exe<<0x011A0002>>
shell.exe<<0x011D0002>>
udevice.exe<<0x01C30002>>
udevice.exe<<0x01F90002>>
udevice.exe<<0x010B0006>>
udevice.exe<<0x02EF0002>>
explorer.exe<<0x038B0002>>
  explorer.exe <<0x038C0002>>
  ?ThreadProc@FileChangeManager@@KAKPAX
  ?CreateTaskBar@@YAKPAX@Z <<0x0213000A>
  ?FontThread@@YAKPAX@Z <<0x02AF000A>>
servicesStart.exe<<0x038F0002>>
  servicesStart.exe <<0x03900002>>
EmulatorStub.exe<<0x03930002>>
  EmulatorStub.exe <<0x03940002>>
servicesd.exe<<0x039D0002>>
  servicesd.exe <<0x039E0002>>
```

```
[Event Info] Wait For Multiple Objects, INFINITE, fWaitAll = 0, Handles = EdbgIntEvent
[Thread Info] KITLInterruptThread <<0x005B0002>>, nPriority = 130
```

Windows Embedded Compact 7

Kernel Tracker View of Boot

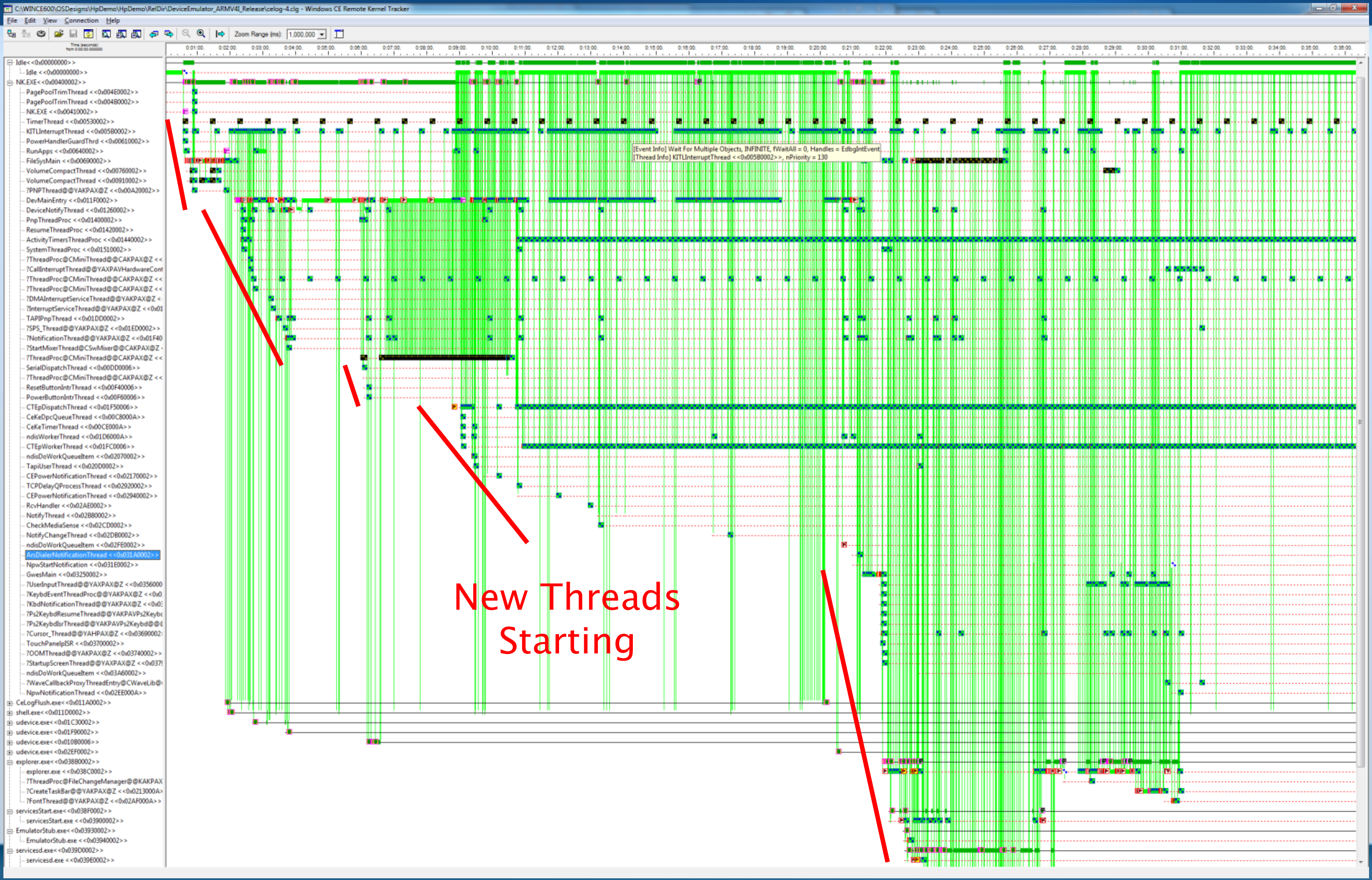


```
Idle<<0x00000000>>
  Idle <<0x00000000>>
NK.EXE<<0x00400002>>
  PagePoolTrimThread <<0x004E0002>>
  PagePoolTrimThread <<0x004B0002>>
  NK.EXE <<0x00410002>>
  TimerThread <<0x00530002>>
  KITLInterruptThread <<0x005B0002>>
  PowerHandlerGuardThrd <<0x00610002>>
  RunApps <<0x00640002>>
  FileSysMain <<0x00690002>>
  VolumeCompactThread <<0x00760002>>
  VolumeCompactThread <<0x00910002>>
  ?PNPThread@@YAKPAX@Z <<0x00A20002>>
  DevMainEntry <<0x011F0002>>
  DeviceNotifyThread <<0x01260002>>
  PnpThreadProc <<0x01400002>>
  ResumeThreadProc <<0x01420002>>
  ActivityTimersThreadProc <<0x01440002>>
  SystemThreadProc <<0x01510002>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?CallInterruptThread@@YAXPAVHardwareCont
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?DMAInterruptServiceThread@@YAKPAX@Z <
  ?InterruptServiceThread@@YAKPAX@Z <<0x01
  TAPIPnpThread <<0x01DD0002>>
  ?SPS_Thread@@YAKPAX@Z <<0x01ED0002>>
  ?NotificationThread@@YAKPAX@Z <<0x01F40
  ?StartMixerThread@CSwMixer@@CAKPAX@Z
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  SerialDispatchThread <<0x00DD0006>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ResetButtonIntrThread <<0x00F40006>>
  PowerButtonIntrThread <<0x00F60006>>
  CTEpDispatchThread <<0x01F50006>>
  CeKeDpcQueueThread <<0x00C8000A>>
  CeKeTimerThread <<0x00CE000A>>
  ndisWorkerThread <<0x01D6000A>>
  CTEpWorkerThread <<0x01FC0006>>
  ndisDoWorkQueueItem <<0x02070002>>
  TapiUserThread <<0x020D0002>>
  CEPowerNotificationThread <<0x02170002>>
  TCPDelayQProcessThread <<0x02920002>>
  CEPowerNotificationThread <<0x02940002>>
  RcvHandler <<0x02AE0002>>
  NotifyThread <<0x02B80002>>
  CheckMediaSense <<0x02CD0002>>
  NotifyChangeThread <<0x02DB0002>>
  ndisDoWorkQueueItem <<0x02FE0002>>
  ArsDialerNotificationThread <<0x031A0002>>
  NpwStartNotification <<0x031E0002>>
  GwesMain <<0x03250002>>
  ?UserInputThread@@YAXPAX@Z <<0x0356000
  ?KeybdEventThreadProc@@YAKPAX@Z <<0x0
  ?KbdNotificationThread@@YAKPAX@Z <<0x0
  ?Ps2KeybdResumeThread@@YAKPAVPs2Keybc
  ?Ps2KeybdIsrThread@@YAKPAVPs2Keybd@@
  ?Cursor_Thread@@YAHPAX@Z <<0x03690002
  TouchPanelpISR <<0x03700002>>
  ?OOMThread@@YAKPAX@Z <<0x03740002>>
  ?StartupScreenThread@@YAXPAX@Z <<0x037
  ndisDoWorkQueueItem <<0x03A60002>>
  ?WaveCallbackProxyThreadEntry@CWaveLib@
  NpwNotificationThread <<0x02EE000A>>
CeLogFlush.exe<<0x011A0002>>
shell.exe<<0x011D0002>>
udevice.exe<<0x01F90002>>
udevice.exe<<0x010B0006>>
udevice.exe<<0x02EF0002>>
explorer.exe<<0x038B0002>>
  explorer.exe <<0x038C0002>>
  ?ThreadProc@FileChangeManager@@KAKPAX
  ?CreateTaskBar@@YAKPAX@Z <<0x0213000A>
  ?FontThread@@YAKPAX@Z <<0x02AF000A>>
servicesStart.exe<<0x038F0002>>
  servicesStart.exe <<0x03900002>>
EmulatorStub.exe<<0x03930002>>
  EmulatorStub.exe <<0x03940002>>
servicesd.exe<<0x039D0002>>
  servicesd.exe <<0x039E0002>>
```

```
[Event Info] Wait For Multiple Objects, INFINITE, fWaitAll = 0, Handles = EdbgIntEvent
[Thread Info] KITLInterruptThread <<0x005B0002>>, nPriority = 130
```

Windows Embedded Compact 7

Kernel Tracker View of Boot



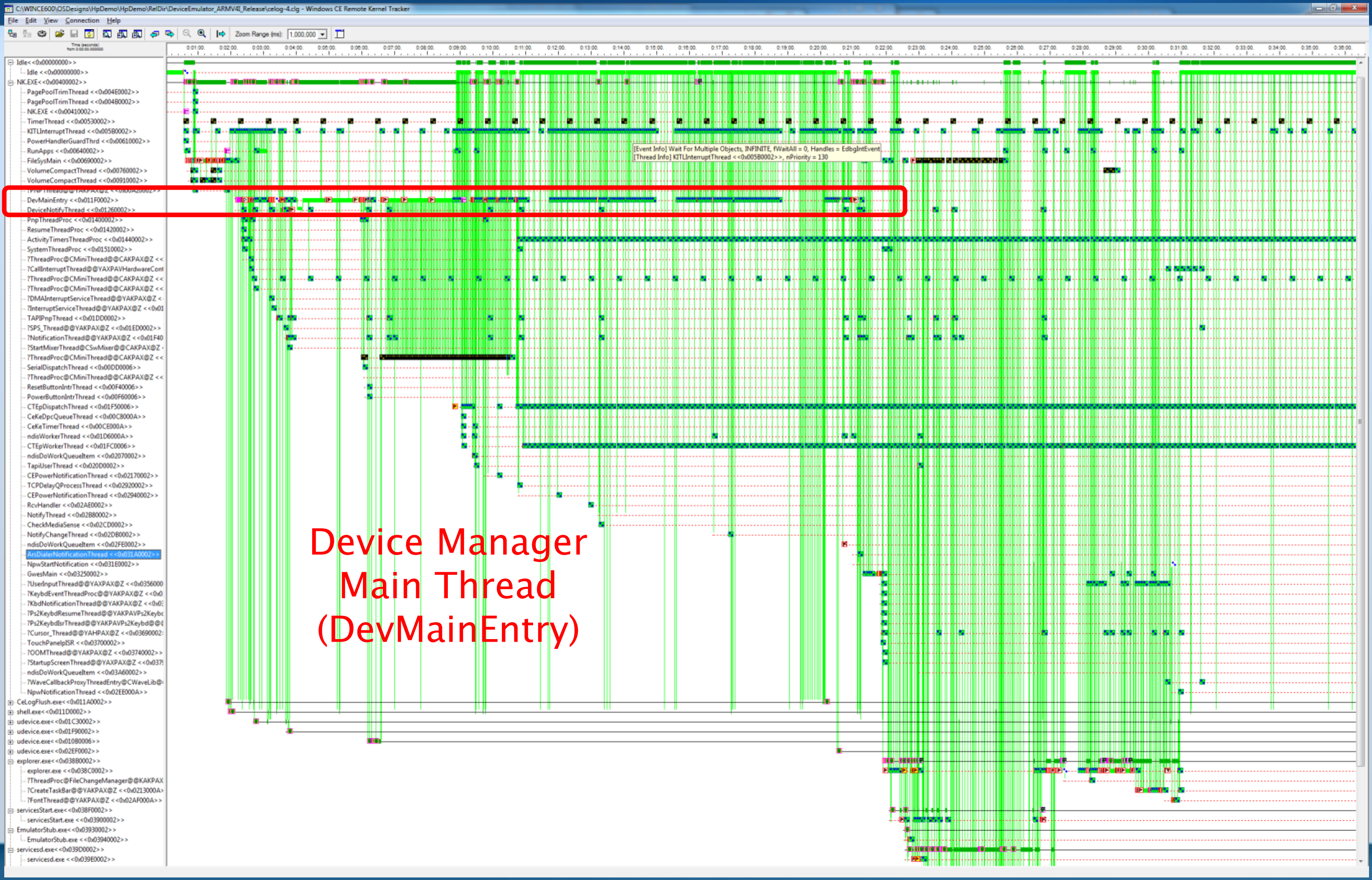
C:\WINCE600\OSDesigns\HpDemo\HpDemo\RelDir\DeviceEmulator_ARMV4I_Release\celog-4.clg - Windows CE Remote Kernel Tracker

Zoom Range (ms): 1,000,000

```
⊟ Idle<<0x00000000>>
    └ Idle <<0x00000000>>
⊟ NK.EXE<<0x00400002>>
    ├ PagePoolTrimThread <<0x004E0002>>
    ├ PagePoolTrimThread <<0x004B0002>>
    ├ NK.EXE <<0x00410002>>
    ├ TimerThread <<0x00530002>>
    ├ KITLInterruptThread <<0x005B0002>>
    ├ PowerHandlerGuardThrd <<0x00610002>>
    ├ RunApps <<0x00640002>>
    ├ FileSysMain <<0x00690002>>
    ├ VolumeCompactThread <<0x00760002>>
    ├ VolumeCompactThread <<0x00910002>>
    ├ ?PNPThread@@YAKPAX@Z <<0x00A20002>>
    ├ DevMainEntry <<0x011F0002>>
    ├ DeviceNotifyThread <<0x01260002>>
    ├ PnpThreadProc <<0x01400002>>
    ├ ResumeThreadProc <<0x01420002>>
    ├ ActivityTimersThreadProc <<0x01440002>>
    ├ SystemThreadProc <<0x01510002>>
    ├ ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ├ ?CallInterruptThread@@YAXPAVHardwareCont
    ├ ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ├ ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ├ ?DMAInterruptServiceThread@@YAKPAX@Z <
    ├ ?InterruptServiceThread@@YAKPAX@Z <<0x01
    ├ TAPIPnpThread <<0x01DD0002>>
    ├ ?SPS_Thread@@YAKPAX@Z <<0x01ED0002>>
    ├ ?NotificationThread@@YAKPAX@Z <<0x01F40
    ├ ?StartMixerThread@CSwMixer@@CAKPAX@Z
    ├ ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ├ SerialDispatchThread <<0x00DD0006>>
    ├ ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ├ ResetButtonIntrThread <<0x00F40006>>
    ├ PowerButtonIntrThread <<0x00F60006>>
    ├ CTEpDispatchThread <<0x01F50006>>
    ├ CeKeDpcQueueThread <<0x00C8000A>>
    ├ CeKeTimerThread <<0x00CE000A>>
    ├ ndisWorkerThread <<0x01D6000A>>
    ├ CTEpWorkerThread <<0x01FC0006>>
    ├ ndisDoWorkQueueItem <<0x02070002>>
    ├ TapiUserThread <<0x020D0002>>
    ├ CEPowerNotificationThread <<0x02170002>>
    ├ TCPDelayQProcessThread <<0x02920002>>
    ├ CEPowerNotificationThread <<0x02940002>>
    ├ RcvHandler <<0x02AE0002>>
    ├ NotifyThread <<0x02B80002>>
    ├ CheckMediaSense <<0x02CD0002>>
    ├ NotifyChangeThread <<0x02DB0002>>
    ├ ndisDoWorkQueueItem <<0x02FE0002>>
    ├ ArsDialerNotificationThread <<0x031A0002>>
    ├ NpwStartNotification <<0x031E0002>>
    ├ GwesMain <<0x03250002>>
    ├ ?UserInputThread@@YAXPAX@Z <<0x0356000
    ├ ?KeybdEventThreadProc@@YAKPAX@Z <<0x0
    ├ ?KbdNotificationThread@@YAKPAX@Z <<0x0
    ├ ?Ps2KeybdResumeThread@@YAKPAVPs2Keybd
    ├ ?Ps2KeybdIsrThread@@YAKPAVPs2Keybd@@
    ├ ?Cursor_Thread@@YAHPAX@Z <<0x03690002
    ├ TouchPanelpISR <<0x03700002>>
    ├ ?OOMThread@@YAKPAX@Z <<0x03740002>>
    ├ ?StartupScreenThread@@YAXPAX@Z <<0x037
    ├ ndisDoWorkQueueItem <<0x03A60002>>
    ├ ?WaveCallbackProxyThreadEntry@CWaveLib@
    └ NpwNotificationThread <<0x02EE000A>>
⊞ CeLogFlush.exe<<0x011A0002>>
⊞ shell.exe<<0x011D0002>>
⊞ udevice.exe<<0x01C30002>>
⊞ udevice.exe<<0x01F90002>>
⊞ udevice.exe<<0x010B0006>>
⊞ udevice.exe<<0x02EF0002>>
⊟ explorer.exe<<0x038B0002>>
    ├ explorer.exe <<0x038C0002>>
    ├ ?ThreadProc@FileChangeManager@@KAKPAX
    ├ ?CreateTaskBar@@YAKPAX@Z <<0x0213000A>
    └ ?FontThread@@YAKPAX@Z <<0x02AF000A>>
⊟ servicesStart.exe<<0x038F0002>>
    └ servicesStart.exe <<0x03900002>>
⊟ EmulatorStub.exe<<0x03930002>>
    └ EmulatorStub.exe <<0x03940002>>
⊟ servicesd.exe<<0x039D0002>>
    └ servicesd.exe <<0x039E0002>>
```

[Event Info] Wait For Multiple Objects, INFINITE, fWaitAll = 0, Handles = EdbgIntEvent
[Thread Info] KITLInterruptThread <<0x005B0002>>, nPriority = 130

Kernel Tracker View of Boot



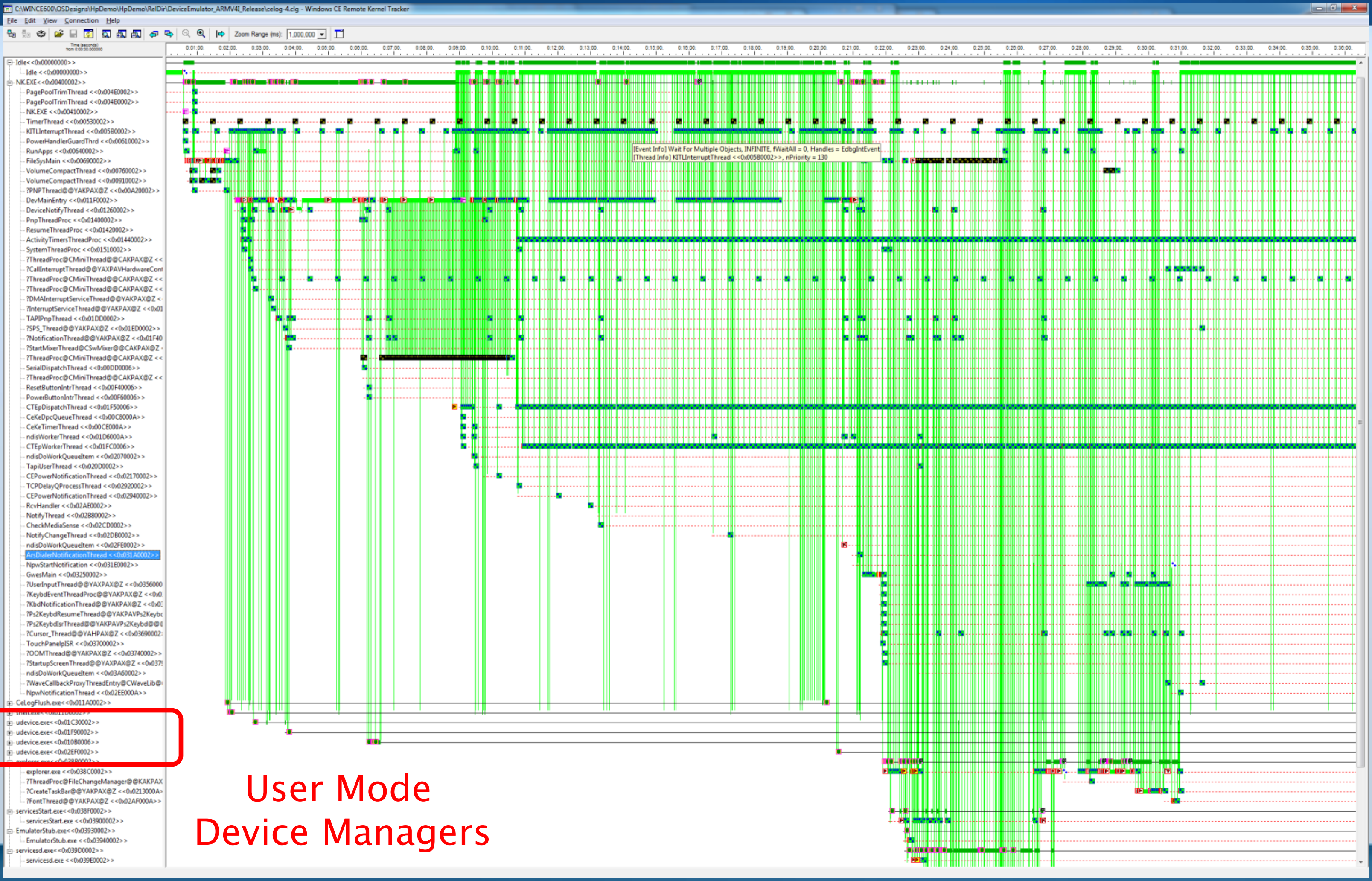
```
Idle<<0x00000000>>
  Idle <<0x00000000>>
NK.EXE<<0x00400002>>
  PagePoolTrimThread <<0x004E0002>>
  PagePoolTrimThread <<0x004B0002>>
  NK.EXE <<0x00410002>>
  TimerThread <<0x00530002>>
  KITLInterruptThread <<0x005B0002>>
  PowerHandlerGuardThrd <<0x00610002>>
  RunApps <<0x00640002>>
  FileSysMain <<0x00690002>>
  VolumeCompactThread <<0x00760002>>
  VolumeCompactThread <<0x00910002>>
  ?PNPThread@@YAKPAX@Z <<0x00A20002>>
  DevMainEntry <<0x011F0002>>
  DeviceNotifyThread <<0x01260002>>
  PnpThreadProc <<0x01400002>>
  ResumeThreadProc <<0x01420002>>
  ActivityTimersThreadProc <<0x01440002>>
  SystemThreadProc <<0x01510002>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?CallInterruptThread@@YAXPAVHardwareCont
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?DMAInterruptServiceThread@@YAKPAX@Z <
  ?InterruptServiceThread@@YAKPAX@Z <<0x01
  TAPIPnpThread <<0x01DD0002>>
  ?SPS_Thread@@YAKPAX@Z <<0x01ED0002>>
  ?NotificationThread@@YAKPAX@Z <<0x01F40
  ?StartMixerThread@CSwMixer@@CAKPAX@Z
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  SerialDispatchThread <<0x00DD0006>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ResetButtonIntrThread <<0x00F40006>>
  PowerButtonIntrThread <<0x00F60006>>
  CTEpDispatchThread <<0x01F50006>>
  CeKeDpcQueueThread <<0x00C8000A>>
  CeKeTimerThread <<0x00CE000A>>
  ndisWorkerThread <<0x01D6000A>>
  CTEpWorkerThread <<0x01FC0006>>
  ndisDoWorkQueueItem <<0x02070002>>
  TapiUserThread <<0x020D0002>>
  CEPowerNotificationThread <<0x02170002>>
  TCPDelayQProcessThread <<0x02920002>>
  CEPowerNotificationThread <<0x02940002>>
  RcvHandler <<0x02AE0002>>
  NotifyThread <<0x02B80002>>
  CheckMediaSense <<0x02CD0002>>
  NotifyChangeThread <<0x02DB0002>>
  ndisDoWorkQueueItem <<0x02FE0002>>
  ArsDialerNotificationThread <<0x031A0002>>
  NpwStartNotification <<0x031E0002>>
  GwesMain <<0x03250002>>
  ?UserInputThread@@YAXPAX@Z <<0x0356000
  ?KeybdEventThreadProc@@YAKPAX@Z <<0x0
  ?KbdNotificationThread@@YAKPAX@Z <<0x0
  ?Ps2KeybdResumeThread@@YAKPAVPs2Keybo
  ?Ps2KeybdIsrThread@@YAKPAVPs2Keybd@@
  ?Cursor_Thread@@YAHPAX@Z <<0x03690002
  TouchPanelpISR <<0x03700002>>
  ?OOMThread@@YAKPAX@Z <<0x03740002>>
  ?StartupScreenThread@@YAXPAX@Z <<0x037
  ndisDoWorkQueueItem <<0x03A60002>>
  ?WaveCallbackProxyThreadEntry@CWaveLib@
  NpwNotificationThread <<0x02EE000A>>
CeLogFlush.exe<<0x011A0002>>
shell.exe<<0x011D0002>>
udevice.exe<<0x01C30002>>
udevice.exe<<0x01F90002>>
udevice.exe<<0x010B0006>>
udevice.exe<<0x02EF0002>>
explorer.exe<<0x038B0002>>
  explorer.exe <<0x038C0002>>
  ?ThreadProc@FileChangeManager@@KAKPAX
  ?CreateTaskBar@@YAKPAX@Z <<0x0213000A>
  ?FontThread@@YAKPAX@Z <<0x02AF000A>>
servicesStart.exe<<0x038F0002>>
  servicesStart.exe <<0x03900002>>
EmulatorStub.exe<<0x03930002>>
  EmulatorStub.exe <<0x03940002>>
servicesd.exe<<0x039D0002>>
  servicesd.exe <<0x039E0002>>
```

```
[Event Info] Wait For Multiple Objects, INFINITE, fWaitAll = 0, Handles = EdbgIntEvent
[Thread Info] KITLInterruptThread <<0x005B0002>>, nPriority = 130
```

Device Manager
Main Thread
(DevMainEntry)

Windows Embedded Compact 7

Kernel Tracker View of Boot



C:\WINCE600\OSDesigns\HpDemo\HpDemo\RelDir\DeviceEmulator_ARMV4I_Release\celog-4.clg - Windows CE Remote Kernel Tracker

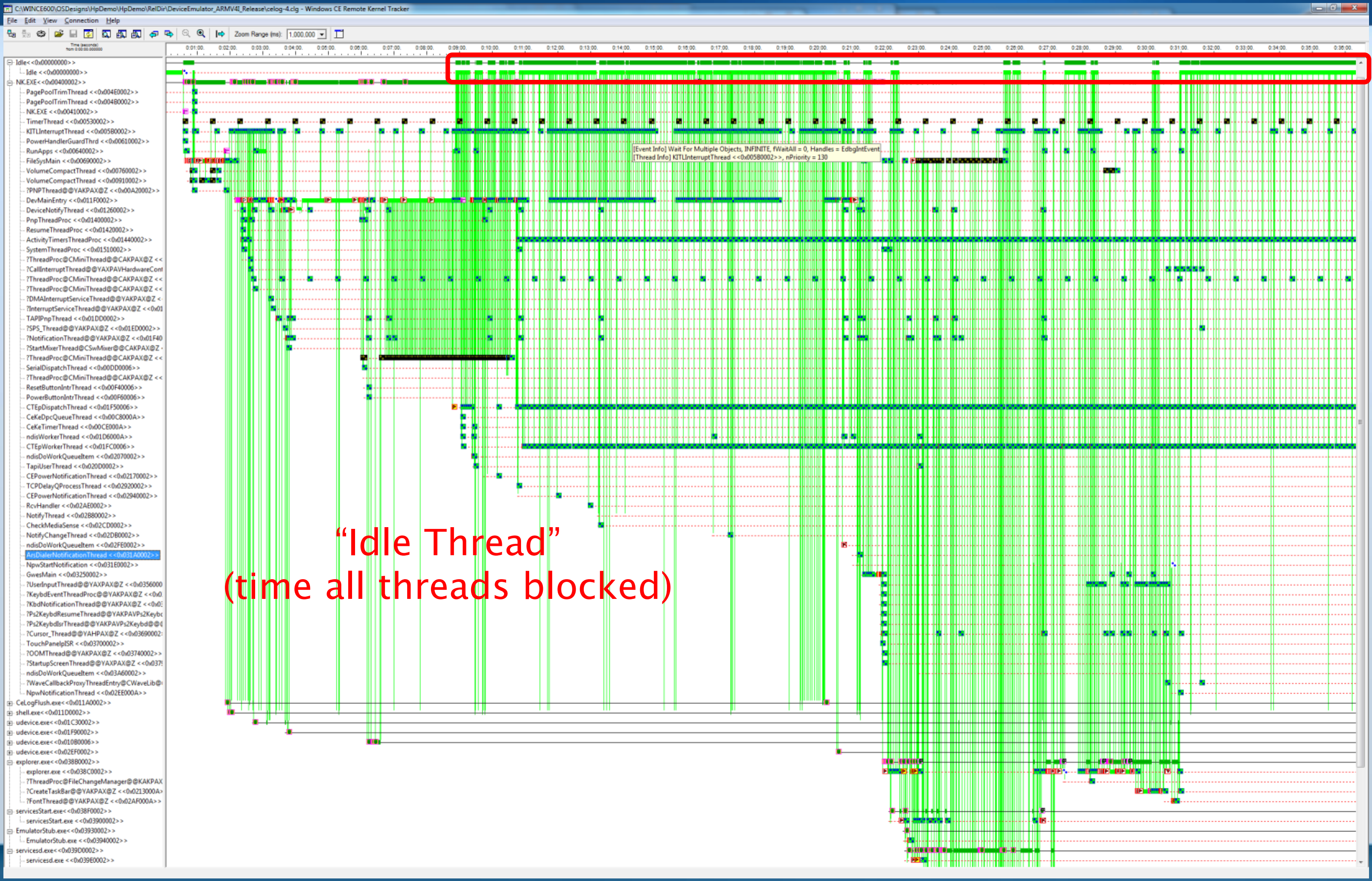
```
Idle<<0x00000000>>
  Idle <<0x00000000>>
NK.EXE<<0x00400002>>
  PagePoolTrimThread <<0x004E0002>>
  PagePoolTrimThread <<0x004B0002>>
  NK.EXE <<0x00410002>>
  TimerThread <<0x00530002>>
  KITLInterruptThread <<0x005B0002>>
  PowerHandlerGuardThrd <<0x00610002>>
  RunApps <<0x00640002>>
  FileSysMain <<0x00690002>>
  VolumeCompactThread <<0x00760002>>
  VolumeCompactThread <<0x00910002>>
  ?PNPThread@@YAKPAX@Z <<0x00A20002>>
  DevMainEntry <<0x011F0002>>
  DeviceNotifyThread <<0x01260002>>
  PnpThreadProc <<0x01400002>>
  ResumeThreadProc <<0x01420002>>
  ActivityTimersThreadProc <<0x01440002>>
  SystemThreadProc <<0x01510002>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?CallInterruptThread@@YAXPAVHardwareCont
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ?DMAInterruptServiceThread@@YAKPAX@Z <
  ?InterruptServiceThread@@YAKPAX@Z <<0x01
  TAPIPnpThread <<0x01DD0002>>
  ?SPS_Thread@@YAKPAX@Z <<0x01ED0002>>
  ?NotificationThread@@YAKPAX@Z <<0x01F40
  ?StartMixerThread@CSwMixer@@CAKPAX@Z
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  SerialDispatchThread <<0x00DD0006>>
  ?ThreadProc@CMiniThread@@CAKPAX@Z <<
  ResetButtonIntrThread <<0x00F40006>>
  PowerButtonIntrThread <<0x00F60006>>
  CTEpDispatchThread <<0x01F50006>>
  CeKeDpcQueueThread <<0x00C8000A>>
  CeKeTimerThread <<0x00CE000A>>
  ndisWorkerThread <<0x01D6000A>>
  CTEpWorkerThread <<0x01FC0006>>
  ndisDoWorkQueueItem <<0x02070002>>
  TapiUserThread <<0x020D0002>>
  CEPowerNotificationThread <<0x02170002>>
  TCPDelayQProcessThread <<0x02920002>>
  CEPowerNotificationThread <<0x02940002>>
  RcvHandler <<0x02AE0002>>
  NotifyThread <<0x02B80002>>
  CheckMediaSense <<0x02CD0002>>
  NotifyChangeThread <<0x02DB0002>>
  ndisDoWorkQueueItem <<0x02FE0002>>
  ArsDialerNotificationThread <<0x031A0002>>
  NpwStartNotification <<0x031E0002>>
  GwesMain <<0x03250002>>
  ?UserInputThread@@YAXPAX@Z <<0x0356000
  ?KeybdEventThreadProc@@YAKPAX@Z <<0x0
  ?KbdNotificationThread@@YAKPAX@Z <<0x0
  ?Ps2KeybdResumeThread@@YAKPAVPs2Keybc
  ?Ps2KeybdIsrThread@@YAKPAVPs2Keybd@@
  ?Cursor_Thread@@YAHPAX@Z <<0x03690002
  TouchPanelpISR <<0x03700002>>
  ?OOMThread@@YAKPAX@Z <<0x03740002>>
  ?StartupScreenThread@@YAXPAX@Z <<0x037
  ndisDoWorkQueueItem <<0x03A60002>>
  ?WaveCallbackProxyThreadEntry@CWaveLib@
  NpwNotificationThread <<0x02EE000A>>
CeLogFlush.exe<<0x011A0002>>
shell.exe<<0x011D0002>>
udevice.exe<<0x01C30002>>
udevice.exe<<0x01F90002>>
udevice.exe<<0x010B0006>>
udevice.exe<<0x02EF0002>>
explorer.exe<<0x038B0002>>
  explorer.exe <<0x038C0002>>
  ?ThreadProc@FileChangeManager@@KAKPAX
  ?CreateTaskBar@@YAKPAX@Z <<0x0213000A>
  ?FontThread@@YAKPAX@Z <<0x02AF000A>>
servicesStart.exe<<0x038F0002>>
  servicesStart.exe <<0x03900002>>
EmulatorStub.exe<<0x03930002>>
  EmulatorStub.exe <<0x03940002>>
servicesd.exe<<0x039D0002>>
  servicesd.exe <<0x039E0002>>
```

```
[Event Info] Wait For Multiple Objects, INFINITE, fWaitAll = 0, Handles = EdbgIntEvent
[Thread Info] KITLInterruptThread <<0x005B0002>>, nPriority = 130
```

User Mode
Device Managers

Windows Embedded Compact 7

Kernel Tracker View of Boot

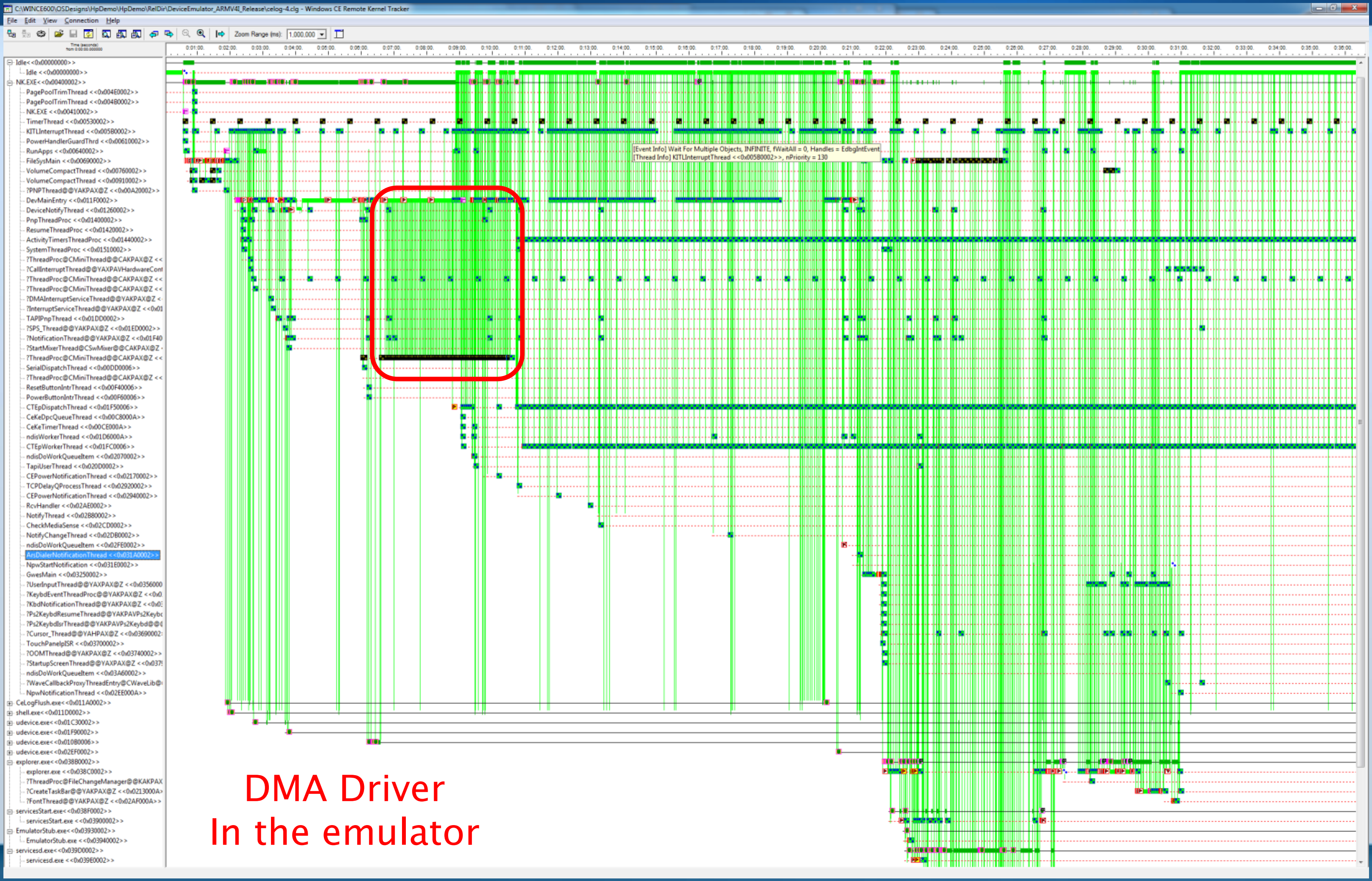

File Edit View Connection Help
Zoom Range (ms): 1,000,000
Time (seconds) from 0:00:00.000000
0:01:00. 0:02:00. 0:03:00. 0:04:00. 0:05:00. 0:06:00. 0:07:00. 0:08:00. 0:09:00. 0:10:00.

```
C:\WINCE600\OSDesigns\HpDemo\HpDemo\RelDir\DeviceEmulator_ARMV4I_Release\celog-4.clg - Windows CE Remote Kernel Tracker

⊟ Idle<<0x00000000>>
    Idle <<0x00000000>>
⊟ NK.EXE<<0x00400002>>
    PagePoolTrimThread <<0x004E0002>>
    PagePoolTrimThread <<0x004B0002>>
    NK.EXE <<0x00410002>>
    TimerThread <<0x00530002>>
    KITLInterruptThread <<0x005B0002>>
    PowerHandlerGuardThrd <<0x00610002>>
    RunApps <<0x00640002>>
    FileSysMain <<0x00690002>>
    VolumeCompactThread <<0x00760002>>
    VolumeCompactThread <<0x00910002>>
    ?PNPThread@@YAKPAX@Z <<0x00A20002>>
    DevMainEntry <<0x011F0002>>
    DeviceNotifyThread <<0x01260002>>
    PnpThreadProc <<0x01400002>>
    ResumeThreadProc <<0x01420002>>
    ActivityTimersThreadProc <<0x01440002>>
    SystemThreadProc <<0x01510002>>
    ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ?CallInterruptThread@@YAXPAVHardwareCont
    ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ?DMAInterruptServiceThread@@YAKPAX@Z <
    ?InterruptServiceThread@@YAKPAX@Z <<0x01
    TAPIPnpThread <<0x01DD0002>>
    ?SPS_Thread@@YAKPAX@Z <<0x01ED0002>>
    ?NotificationThread@@YAKPAX@Z <<0x01F40
    ?StartMixerThread@CSwMixer@@CAKPAX@Z
    ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    SerialDispatchThread <<0x00DD0006>>
    ?ThreadProc@CMiniThread@@CAKPAX@Z <<
    ResetButtonIntrThread <<0x00F40006>>
    PowerButtonIntrThread <<0x00F60006>>
    CTEpDispatchThread <<0x01F50006>>
    CeKeDpcQueueThread <<0x00C8000A>>
    CeKeTimerThread <<0x00CE000A>>
    ndisWorkerThread <<0x01D6000A>>
    CTEpWorkerThread <<0x01FC0006>>
    ndisDoWorkQueueItem <<0x02070002>>
    TapiUserThread <<0x020D0002>>
    CEPowerNotificationThread <<0x02170002>>
    TCPDelayQProcessThread <<0x02920002>>
    CEPowerNotificationThread <<0x02940002>>
    RcvHandler <<0x02AE0002>>
    NotifyThread <<0x02B80002>>
    CheckMediaSense <<0x02CD0002>>
    NotifyChangeThread <<0x02DB0002>>
    ndisDoWorkQueueItem <<0x02FE0002>>
    ArsDialerNotificationThread <<0x031A0002>>
    NpwStartNotification <<0x031E0002>>
    GwesMain <<0x03250002>>
    ?UserInputThread@@YAXPAX@Z <<0x0356000
    ?KeybdEventThreadProc@@YAKPAX@Z <<0x0
    ?KbdNotificationThread@@YAKPAX@Z <<0x0
    ?Ps2KeybdResumeThread@@YAKPAVPs2Keybd
    ?Ps2KeybdIsrThread@@YAKPAVPs2Keybd@@
    ?Cursor_Thread@@YAHPAX@Z <<0x03690002
    TouchPanelpISR <<0x03700002>>
    ?OOMThread@@YAKPAX@Z <<0x03740002>>
    ?StartupScreenThread@@YAXPAX@Z <<0x037
    ndisDoWorkQueueItem <<0x03A60002>>
    ?WaveCallbackProxyThreadEntry@CWaveLib@
    NpwNotificationThread <<0x02EE000A>>
⊞ CeLogFlush.exe<<0x011A0002>>
⊞ shell.exe<<0x011D0002>>
⊞ udevice.exe<<0x01C30002>>
⊞ udevice.exe<<0x01F90002>>
⊞ udevice.exe<<0x010B0006>>
⊞ udevice.exe<<0x02EF0002>>
⊟ explorer.exe<<0x038B0002>>
    explorer.exe <<0x038C0002>>
    ?ThreadProc@FileChangeManager@@KAKPAX
    ?CreateTaskBar@@YAKPAX@Z <<0x0213000A>
    ?FontThread@@YAKPAX@Z <<0x02AF000A>>
⊟ servicesStart.exe<<0x038F0002>>
    servicesStart.exe <<0x03900002>>
⊟ EmulatorStub.exe<<0x03930002>>
    EmulatorStub.exe <<0x03940002>>
⊟ servicesd.exe<<0x039D0002>>
    servicesd.exe <<0x039E0002>>

[Event Info] Wait For Multiple Objects, INFINITE, fWaitAll = 0, Handles = EdbgIntEvent
[Thread Info] KITLInterruptThread <<0x005B0002>>, nPriority = 130
```

"Idle Thread"
(time all threads blocked)

Windows Embedded Compact 7

Kernel Tracker View of Boot


C:\WINCE600\OSDesigns\HpDemo\HpDemo\RelDir\DeviceEmulator_ARMV4I_Release\celog-4.clg - Windows CE Remote Kernel Tracker
File
Edit
View
Connection
Help
Zoom Range (ms): 1,000,000
Idle<<0x00000000>>
Idle <<0x00000000>>
NK.EXE<<0x00400002>>
PagePoolTrimThread <<0x004E0002>>
DMA Driver
In the emulator

[Event Info] Wait For Multiple Objects, INFINITE, fWaitAll = 0, Handles = EdbgIntEvent
[Thread Info] KITLInterruptThread <<0x005B0002>>, nPriority = 130

Windows Embedded Compact 7

Inserting Custom Data in CeLog

- This API logs data

```
void CeLogData (BOOL fTimeStamp, WORD wID, PVOID pData,
           WORD wLen, WORD dwZoneUser, DWORD dwZoneCE,
           WORD wFlag, BOOL fFlagged);
```

– fTimeStamp	TRUE to add timestamp to entry
– wID	Log ID – See next slide
– pData	Pointer to data to log
– wLen	Length of data
– dwZoneUser	User defined zones
– dwZoneCE	Zone the event relates to
– wFlag	User defined flag
– fFlagged	TRUE to logging wFlag field

Windows Embedded Compact 7

Inserting Custom Data in CeLog

- Predefined data types
  - Each predefined data type can log an array of that type
  - Character   unsigned character   wide char
  - Short   unsigned short
  - Long   unsigned long
  - Float   double
- Custom types can be logged as well using IDs ranging from CELID_USER to CELID_MAX
  - All are defined in ..\public\common\oak\sdk\celog.h
  - Custom types can be interpreted using ReadLog extensions

Boot Time Tips

Windows Embedded Compact 7

Tune Machine Startup

- Disable memory tests unless needed
- Hide BIOS / EFI messages
  - The user doesn’t need to see the PCI device enumeration
- Disable floppy and other disk checks
- If using BIOS / EFI, extend to add splash screen

Tune the Loader Code

- Bootloaders typically copy the .bin file from storage to RAM
  - Optimizing this copy can shave seconds off the boot
  - Look at hardware interface to optimize read from flash
- Keep the image as small as possible
  - Remove unneeded components
  - Consider breaking the .bin file into parts
    - Multiple bin files or a single bin file and discreet files in the file system
- Display a splash screen with a progress bar as quickly as possible.
  - If possible, design OAL so splash screen remains until display driver up

Smaller Images

- Smaller images are better images
  - Faster to load a small image than a large one
  - Less code means smaller RAM footprint
  - Less ‘black box’ code doing things you don’t know about

- Break up the image if necessary
  - Balance boot speed requirement with engineering resources
    - Understanding of the build process
    - Need to package all parts of image and deliver it to device
    - Develop an update strategy

Optimize the Driver Initialization

- Driver loading is a major component of the boot process
- Remove unneeded drivers
  - Do you need all drivers in shipping version?
- Group user mode drivers in one or two UM Driver Managers
  - By default each UM driver gets its own process
  - The more processes that start, the longer the boot takes

Driver Init Procedure Optimization

- Driver Init procedures are called serially during boot
  - A single driver can slow down the boot
- Put Interrupt Service Thread initialization in that thread
  - IST should read its own registry entries, set its own priorities and such
- Don’t wait on hardware
  - Use another thread to wait on the hardware
  - Have the driver fail open calls until hardware is ready

Only Load the Services You Need

- Many services are added by the default configurations
  - OBEX
  - TimeService
- Unless you need a specific service, don’t use it
- If all services can be eliminated remove the services manager
  - If you need one ‘service’ consider writing it as a driver

Remove Explorer unless absolutely needed

- The Explorer is very useful during bring up
  - Much less so when the system has shipped
- While pretty quick, it does take time to launch
  - Save time, eliminate it
- Launch apps on boot using registry
  - Use Init key instead of Explorer startup folder
- Frees custom application to handle “system keys”
  - “Windows” key combinations and select Alt-key combinations

Manage application startup

- Don’t install the application on cold boot
  - Don’t laugh, I’ve seen this!
- Use registry initialization file to provide needed registry keys
  - You’ll need to teach the application developers how to do this
- Use a custom .bib file to allow application to prepopulate files
  - Or provide a method to prepopulate file on storage device

Other Thoughts…

- RAM based registry is much faster than Hive based registry
- May need to consider suspend / resume if system too big
  - Yes, the operating system still supports this (quite well actually)
- Consider Hibernate
  - Suspend with RAM saved on storage device
  - No Microsoft provided code but fairly easy conceptually
- Don’t expose technical boot messages to user
  - Think what your Mom would like to see

Summary

- Boot time has a huge impact of “First Impression”
- Every second of every boot of every device…

  … can save “Lives” of time.
- Use CeLog
  - The best source for boot time information
- Get it right

Questions…

Doug Boling

Boling Consulting Inc.

www.bolingconsulting.com

dboling @ bolingconsulting.com

Microsoft®