

# Optimizing Your Image Size in Windows Embedded Compact 7

Douglas Boling  
Boling Consulting Inc.

# 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

- Image size basics
- Big Mammals
- Using Platform Builder to understand Image configuration
- Changes in Windows Embedded Compact 7
- Impact on software compatibility

# Why Do You Care?

- Boot speed
  - Smaller systems boot faster
- Stability
  - No bugs in code not included in platform
  - Security footprint
- Hardware cost
  - Smaller images mean
    - Less Flash
    - Less RAM

# What Does it Cost?

- Engineering time
  - It's easy to simply "Click and Build"
  - It takes time to shrink the OS the right way
- Compatibility
  - You may remove a component needed by an application

# CPU Differences

- Different CPU architectures have different code densities
  - Dependent on op code efficiency
  - Op code alignment requirements
- x86 provides smallest code
- ARM approximately 1.5 x bigger
  - Thumb helps shrink code

# Changes in Windows Embedded Compact 7

- Increased functionality
  - Crypto API
- Increased dependencies
  - Explorer Shell
  - ActiveSync
  - Media Player
- XAML dependencies
  - New set of parallel components that depend on XAML

# Changes in Windows Embedded Compact 7

- Microsoft changed the linker settings in 7
  - Goal was performance improvement
- Section size changed from 512 to 4096
  - This forces the section boundaries to align with the page size
- Causes .EXE and .DLL files to significantly grow
  - Most of this space is recovered if module in .BIN file
  - However, if loaded in file system, the file is bigger
  - What this component needs
  - Who needs this component



# Componentization

- Windows Embedded Compact 7 is a series of components
  - Some binary
  - Some in source
- Componentization at the object file level
- Platform Builder analyzes components and adds dependencies
  - It's the dependencies that bloat the size!
- Beware the API impact
  - Reducing the OS removes various APIs

# Size Comparisons

- Thin Client 27,364 LB
  - With .NET CF, Explorer Shell, IE for Embedded
- Thin Client Removing IE for Embedded 22,118 KB
- Thin Client removing Explorer Shell 19,225 KB
- Thin Client removing .NET CF 16,614 KB
- Basic OS with no shell 6,226 KB
  - Includes GWE, FileSys, Kernel

# Big Mammals

- Flash 6,195 KB
- Internet Explorer for Embedded 5,246 KB
- Silverlight 4,535 KB
- .NET CF 3.5 2,611 KB
- SQL 2,508 KB

# Embedded Database Component

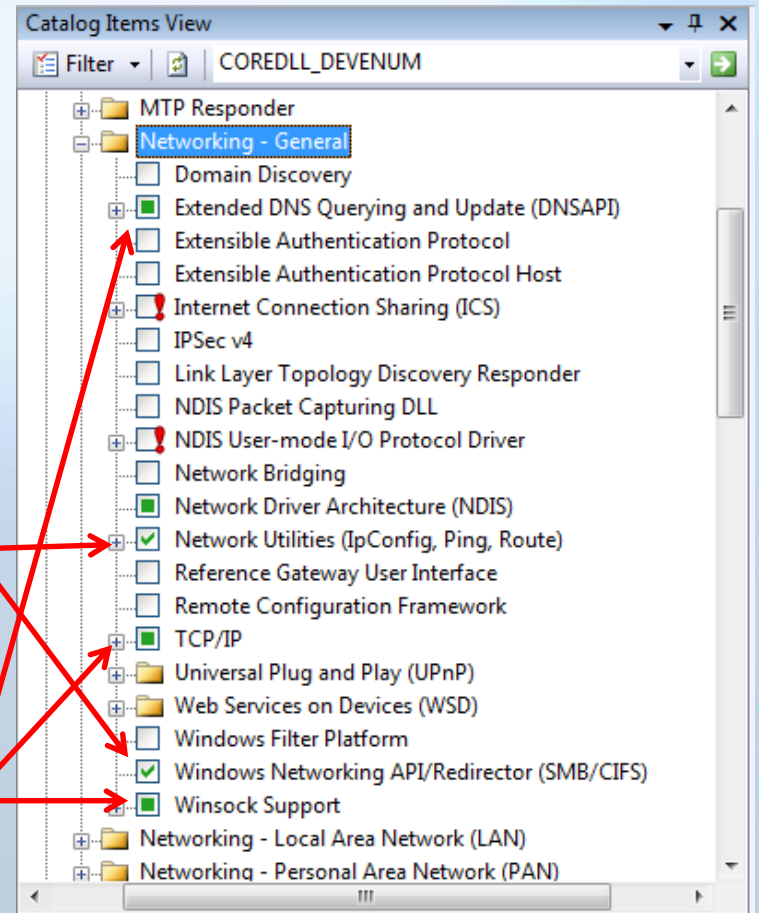
- Embedded Database an 'upgrade' to the old CE database
  - More sort features among others
- Pulls in the basic SQL engine
  - sqlcese35.sys.dll                      496,640 KB
- Many unexpected components 'depend' on EDB
  - ActiveSync, Media Library, POOM, Windows Music Player
- If specifically included...
  - Do you really need the extra features of the EDB?

# Dependencies

- Platform Builder adds dependency components based on the components added by the developer

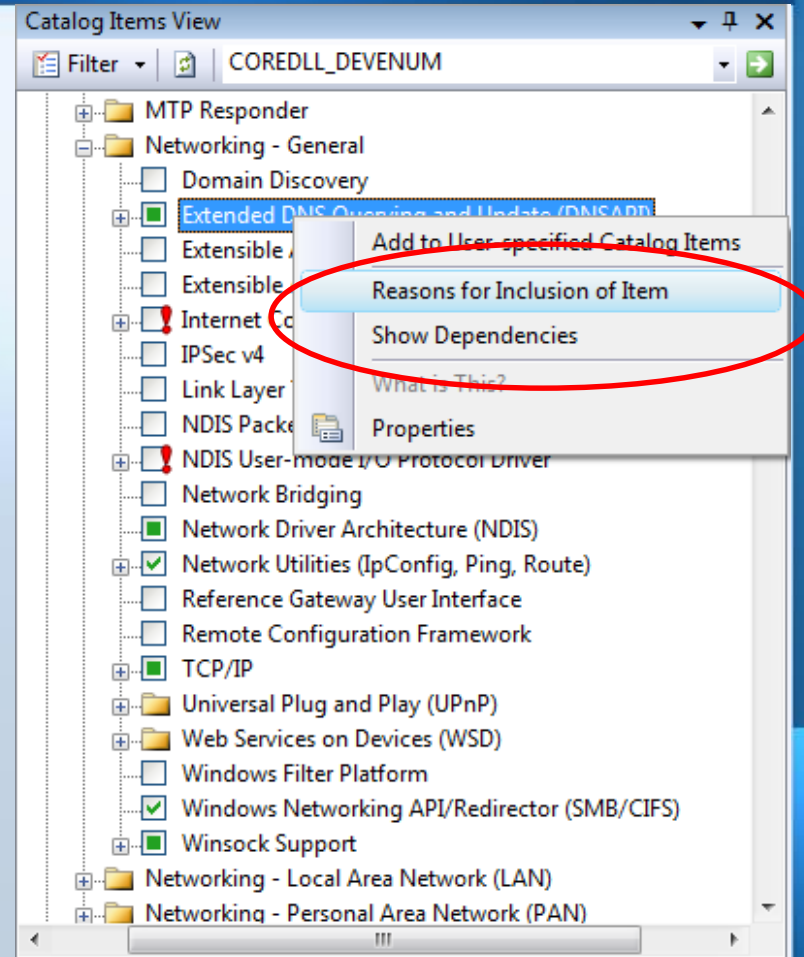
User selected components

PB added dependencies



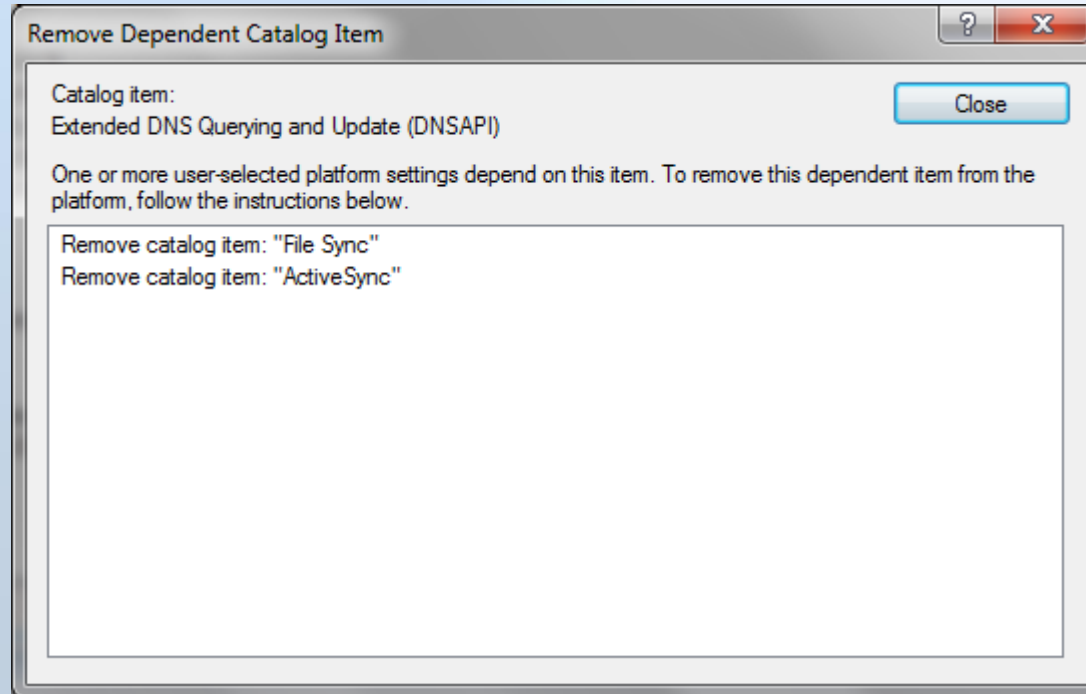
# Diagnosing Dependencies

- Right click on an item to determine why it is included
- Can also show dependencies



# Reasons for Inclusion

- Dialog showing what components caused item to be included
  - Remove those components and this item will be removed



# Know about cebasecesysgen.bat

- BAT file that is in the root of your BSP directory
  - May call other BAT files
- Allows the BSP to change the SYSGEN settings
  - Generally adding components
  - Wrong in so many ways
- Some silicon vendors use this to force OS features that show off their silicon



# Other Tools and Techniques

- Opening NK.BIN with Platform Builder
  - ViewBin tool
- CE.BIB for files being included
- Coredll.def for APIs in a component

# Computing the Size Impact of a Component

- Get SYSGEN tag from properties of catalog item
- Go to project that contains those items
  - This may take some searching in wince700\public
- Find the .bib file in <proj>\oak\files\<proj>.bib
- Look at the files listed within the CESYSGEN tags
- Find their size in the release directory

# Software Compatibility

- It's important to understand the impact of shrinking the OS
  - APIs disappear
- Software implicitly linking to removed APIs won't load
- Particular problem with 3<sup>rd</sup> party binaries
  - If you can't recompile to solve the problem

# Other Solutions to the Image Size Problem

- If the image size is important for boot speed
  - Consider multiple .BIN files for booting
- Will a headless system provide enough functionality
  - Beware of compatibility
- Is this just because Debug builds won't fit?

# Headless Systems

- A headless system is significantly smaller
  - But lots of code uses a subset of GWES
- Sometimes its best to include GWES with a null display
  - Yes, the system will be bigger
- Then componentize everything else out related to U/I
  - Imaging, alphablend, and such components

# Debug vs. Checked vs. Release Builds

- Debug builds include nice features
  - Heap checking
  - Better parameter checking
  - Line by line debugging
- Checked builds a good compromise
  - Debugging features but no line by line debugging (optimizations)
- Mix builds by manually copying modules to retail
  - Enables performance of retail with necessary debug-able modules

# Recommendations

- Avoid Explorer shell
- Avoid ActiveSync
- Beware dependencies
- Mix debug/checked/retail components

# Summary

- Size isn't a problem...
  - ...until it is
- The OS is bigger
  - Some of this is due to better functionality
- Some components have significantly more dependencies
  - ActiveSync, Explorer Shell
- Know the components
  - Many you won't need, some you will



# Questions...

Doug Boling

Boling Consulting Inc.

[www.bolingconsulting.com](http://www.bolingconsulting.com)

[dboling @ bolingconsulting.com](mailto:dboling@bolingconsulting.com)

***Microsoft***<sup>®</sup>