Mobile Opportunities for the Open Source Community

Ravi Belwal (ravi.belwal@nokia.com)
Sr. Technology Consultant
Forum Nokia
S60: IT’S WHAT’S IN YOUR PHONE
S60 is the leading converged device platform and application suite

- Complete mobile application suite
- Flexible runtime offering
- Reliable middleware and integration layer

- S60 Application Suite
  - Licensee Applications
  - Operator Applications
  - Developer Applications

- S60 Scalable UI
  - Web Tech.
  - Flash Lite
  - C++/Open C
  - Java
  - Other Runtimes
  - Python

- Platform Libraries and middleware

- Symbian OS and Security Platform

- Hardware (incl. Security and Multi-radio)
S60 runs over 100 million phones around the world.

>50% share of worldwide converged device market

S60 licensees
Over 50 S60 models sold by 200+ operators
The **widest choice** of runtime technologies provides **maximum flexibility** to developers

**S60** has a solution that meets your requirements for features, user experience, and portability.

- **Powerful, native** technology for expert developers to extend the platform and create custom solutions.
- **Practical, portable** ways to deploy components and applications across different devices and platforms.
- **Easy-to-program, efficient** ways to create exciting user experiences for applications and services.
- **Open C**, **Java**, **C++**, **Flash**, **Python**, **Web Run-Time**
Introduction to Open C

- Typical Application Architecture
- S60 Platform Porting Problem
- Open C And Its Benefits
Typical Application Architecture

- Applications split into 2 conceptual parts:
  - GUI module
  - Engine module
- Implementation
  - GUI and Engine in a Single exe (small apps)
  - Engine in a separate DLL (typical)
  - A number of binary components (large and complex applications)

![Diagram showing the typical application architecture with clearly defined interfaces between GUI and Engine modules.]
S60 Platform Porting Problem

• Porting to many platforms:
  • GUI needs rewriting but Engine requires minimal changes

• Porting to S60 Platform:
  • GUI needs rewriting and Engine requires major rework
  • **In many cases this is too much effort to port to S60!**

• Can now use Open C
  • Allows Engine to be ported with much less effort
    • Provided it is C based code using POSIX/glib libraries
  • Only for Engine code though!
    • GUI must still be written in the standard Symbian way
Open C leverages the flexibility of open source to reduce development costs

- Reduce time to market by taking advantage of existing components and open source projects
- Reduce development costs by using a common codebase across platforms
- Reduce project startup time by drawing from a larger pool of qualified developers

Share components among platforms

Other Symbian platforms  Linux/UNIX

Open C

UI

Runtime technologies

Symbian OS
## Open C Coverage

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
<th>Open Source Project</th>
<th>Coverage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>libc</td>
<td>Standard C libraries</td>
<td>POSIX</td>
<td>47</td>
</tr>
<tr>
<td>libpthread</td>
<td>Implements multi-threading within a traditional user process.</td>
<td>POSIX</td>
<td>60</td>
</tr>
<tr>
<td>libm</td>
<td>Arithmetical and mathematical functions</td>
<td>POSIX</td>
<td>42</td>
</tr>
<tr>
<td>libdl</td>
<td>Allows loading of dynamic link libraries.</td>
<td>POSIX</td>
<td>100</td>
</tr>
<tr>
<td>libz</td>
<td>Provides in-memory compression and decompression functionality.</td>
<td>LIBZ</td>
<td>100</td>
</tr>
<tr>
<td>libcrypt</td>
<td>Provides functionality for encrypting blocks of data, messages, and password hashing.</td>
<td>OpenSSL</td>
<td>100</td>
</tr>
<tr>
<td>libcrypto</td>
<td>Implements a wide range of cryptographic algorithms used in various Internet standards.</td>
<td>OpenSSL</td>
<td>77</td>
</tr>
<tr>
<td>libglib</td>
<td>A general-purpose utility library.</td>
<td>GNOME</td>
<td>100</td>
</tr>
<tr>
<td>libssl</td>
<td>Implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols.</td>
<td>OpenSSL</td>
<td>86</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>72</strong></td>
</tr>
</tbody>
</table>

*Percentage of functions of the full open-source project that is included in Open C.*
Open C Offering

• For S60 3rd Edition and For S60 3rd Edition FP1 SDKs
  • Open C comes as a plugin to install over the SDK
• For S60 3rd Edition FP2 and beyond
  • Open C will be integrated into the SDK
• All S60 3rd Edition SDKs and Open C plugin are freely downloadable from the Forum Nokia website
• Open C comes with:
  • Open C APIs (headers and libraries)
  • Documentation (Open C API reference and Developer Guides)
  • Example applications
  • Open C runtime for S60 3rd Edition devices
What you need to know about S60?

- How To Program For Mobile Devices?
- Tools
- SDKs
- Carbide.c++
Factors to consider when developing for mobile devices:

- Robustness
  - Device must be able to run for very long periods without rebooting
- Code Size
  - Not so much of a problem now with MMC cards
- Memory Usage
  - Available memory is low compared with PCs and likely to run out
- Performance
  - Processors are slow compared to those on PCs
- Battery Life
  - Minimise processor usage to extend battery life
- Screen Size
  - Devices have limited screen size
Writeable Static data

- Global writeable static data (WSD) is any per-process variable which exists for the lifetime of the process.
  - i.e. Globally scoped data and function scoped static variables
    ```c
    int index; //WSD
    void SetFileName()
    {
        static int count; //WSD
        ...
    }
    ```
- WSD in EXEs is supported by all versions of S60
- WSD in DLLs is supported by S60 3rd Edition BUT
  - Is very memory inefficient
  - Should use only as a last resort! – For example use TLS instead!
  - Porting projects are usually justified in using WSD
- Add `EPOMATICALLOWDLLDATA` keyword to mmp file.
Tools

- An Integrated Development Environment (IDE)
  - Preferably Carbide.c++

- An S60 3rd Edition SDK
  - Contains all the necessary S60 C++ headers and libs
  - Contains a free target gcc compiler (install separately)

- Java Runtime
  - Required by SDK and by Carbide.c++

- Perl
  - Required by SDK and by Carbide.c++

- Open C Plugin
  - Required for S60 3rd Edition and S60 3rd Edition FP1 SDKs only
S60 3rd Edition SDKs

- S60 3rd Edition SDK contents:
  - Emulator – Win32 port of S60 Platform
    - Used for viewing and testing
  - System header files
    - `<EPOCROOT>\epoc32\include`
  - Two sets of library files (for emulator and target)
    - `<EPOCROOT>\epoc32\release`
  - Example Code
    - `<EPOCROOT>\Examples` (Symbian Examples)
    - `<EPOCROOT>\S60Ex` (S60 Examples)
  - Documentation
    - `<EPOCROOT>\S60Doc`
Carbide.c++

- Fully featured IDE based on Eclipse and Java
- Carbide.c++ comes in a number of Editions:
  - Express (free), Developer, Professional, OEM
- Comes with an x86 compiler
  - For building software for the Emulator
Where to go from here?

- Download Tools
  - [www.forum.nokia.com](http://www.forum.nokia.com) for SDKs and Carbide.c++
  - [www.java.com](http://www.java.com) for java runtime
  - [www.activestate.com](http://www.activestate.com) for active perl
- S60 SDK Documentation – *.chm file in S60Doc folder
  - Available as part of any S60 SDK
  - Gives further information on all aspects of S60
- Additional Open C Material
  - Search the forum nokia website ([www.forum.nokia.com](http://www.forum.nokia.com)) for further information on Open C
- Certification
  - For further information on certification browse the Symbian Signed website ([www.symbiansigned.com](http://www.symbiansigned.com))
Thank you!

More information: http://www.forum.nokia.com