The POCO Platform for Embedded Development

Günter Obiltschnig
Applied Informatics Software Engineering GmbH
guenter.obiltschnig@appinf.com

applied informatics

"Without a good library, most interesting tasks are hard to do in C++; but given a good library, almost any task can be made easy."

Bjarne Stroustrup

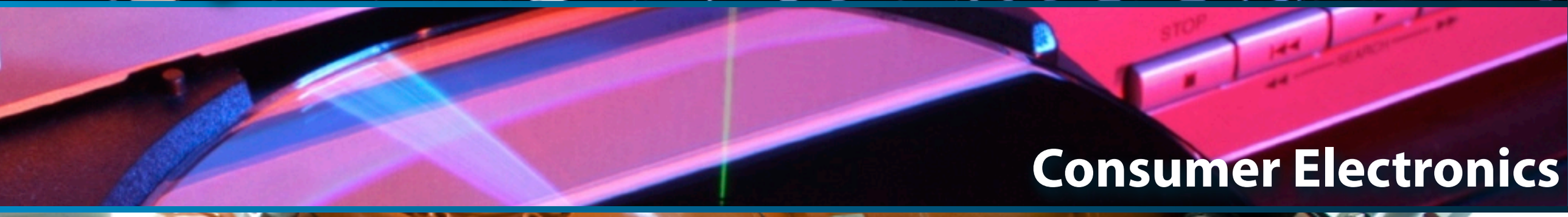
One Software **Platform**

Unlimited Possibilities


Automotive Test and Diagnostic Systems


Building and Home Automation Systems


Ticketing/Entrance Control


Consumer Electronics


Industrial Automation

Applications

Netconf

Zeroconf

Universal Plug and Play

Remoting

Binary

SOAP

Code Generator

OSP

Standard Services

Bundle Creator

Data

SQLite

ODBC

MySQL

Middleware

XML
Fast Infoset

Net
NetSSL

Util

WebWidgets

Foundation

C++ and C Standard Libraries

POSIX, WIN32, other (RT)OS API

Linux Drivers & Kernel Modules

The POCO Platform Consists of:

- The **POCO C++ Libraries**:
  Free general-purpose open source C++ libraries developed by an enthusiastic community of C++ developers.

- **Commercial libraries and tools** built on top of the POCO C++ Libraries providing high-level features, such as remoting and web services, as well as a flexible, component-based module system. Licensed under a commercial source code license (no runtime licenses or royalties).

The POCO C++ Libraries are...

- a collection of C++ class libraries, similar in concept to the Java Class Library, the .NET Framework or Apple's Cocoa;
- focused on "internet-age" network-centric applications;
- written in modern ANSI/ISO Standard C++ and based on the C++ Standard Library/STL;
- highly portable and available on many different platforms;
- Open Source, licensed under the Boost Software License,
- and thus completely free for both commercial and non-commercial use.

POCO Objectives and Mission

- POCO is a powerful, yet easy to use platform to build your applications upon
- POCO allows you to build highly portable applications (write once – compile and run anywhere)
- POCO is modular and scalable from embedded to enterprise applications (you only pay for what you use)
- POCO provides consistent, comprehensive and comprehensible programming interfaces
- POCO is written in fast, efficient C++

Objectives and Mission (cont'd)

- POCO favors simplicity over complexity ("as simple as possible, but not simpler")
- POCO aims for consistency in design, coding style and documentation
- POCO emphasizes source code quality, in terms of readability, comprehensiveness, consistency, style and testability
- POCO aims to make C++ programming fun again

Guiding Principles

- Strong focus on **code quality**, **style**, **consistency** and **code readability** –all code must satisfy our coding styleguide (and it works – we frequently get compliments on our code quality)
- Strong focus on tests (automated unit tests with high coverage)
- Favor pragmatic and elegant design over "solving all the worlds problems" (if we can satisfy 95 % of all use cases with an elegant solution, and the remaining 5 % would require an overly complex design, we focus on the 95 %)
- Build on top of solid foundations – use existing proven C libraries (e.g., expat, zlib, PCRE) where it makes sense

History

- Summer 2004: Günter Obiltschnig started development
- February 2005: First release on SourceForge
  (Release 0.91 under Sleepycat license)
- May 2005: First contributions by Aleksandar Fabijanic
- January 2006: Release 1.0
- March 2006: Release 1.1
- July 2006: Moved to Boost license, POCO Community Website
- August 2006: Release 1.2
- May 2007: Release 1.3
- Early 2009: about 20 contributors, used in 100s of projects

POCO Usage Examples

- building automation middleware and devices
- industrial automation and industrial equipment
- traffic control systems
- healthcare applications
- measurement, data acquisition and test systems
- consumer electronics/home automation
- air traffic management systems
- VoIP
- ticketing and entrance-control systems
- shrink-wrapped applications

Some Companies using POCO

- **454 Life Sciences (Roche)**
  using POCO in its new high-speed genome sequencer
- **ACTIA Automotive**
  using POCO and OSP in automotive diagnostic systems
- **Adobe Systems**
  using POCO in a VoIP platform
- **CACE Technologies**
  using POCO in its Pilot product
- **Comact Optimisation**
  using POCO in sawmill equipment running under QNX

Some Companies using POCO (cont'd)

- **Nucor Steel**
  using POCO (and OSP) in beam mill automation applications
- **R-SYS**
  using POCO in air traffic management systems
- **StreamUnlimited**
  using POCO in embedded applications for set-top boxes
- **Starticket**
  using POCO and Remoting in a ticketing/entrance control system running on an embedded Linux platform
- **TAC**
  using POCO in building automation platform (running on embedded Linux systems and Windows/Linux servers)

POCO – Scalability Embedded

- POCO is well-suited for embedded systems running under Embedded Linux or QNX.
- POCO-based applications (using the built-in web server) run on 75 MHz ARM9-based Linux systems (uClibc) with 8 MB RAM and 4 MB Flash (e.g. Digi Connect ME 9210).
- A typical POCO-based application using the web server from the Net library has a statically linked size of 2 MB and uses about 2 – 3 MB of RAM.

- Typical mid-ranged embedded platforms (32 – 64 MB RAM, 16 – 64 MB Flash, 180 MHz ARM9) provide plenty of resources even for more complex applications (using OSP and Remoting).

POCO and Embedded – Code Size

- POCO (including SSL/Crypto) + OSP + Remoting/NETCONF libraries use less than 4 MB of Flash storage (compressed jffs2 or squashfs).
- The RAM overhead for such an application is below 8 MB.

```
guenter@cis-digiel:~/ws/poco-1.3$ ls -l lib/Linux/armv5tejl/*.so.*
-rwxr-xr-x 1 guenter guenter  103616 2009-03-03 18:48 lib/Linux/armv5tejl/libPocoBinary.so.6
-rwxr-xr-x 1 guenter guenter  103752 2009-03-03 19:12 lib/Linux/armv5tejl/libPocoCrypto.so.6
-rwxr-xr-x 1 guenter guenter 1582720 2009-03-03 18:43 lib/Linux/armv5tejl/libPocoFoundation.so.6
-rwxr-xr-x 1 guenter guenter  466384 2009-03-03 18:49 lib/Linux/armv5tejl/libPocoNetconf.so.6
-rwxr-xr-x 1 guenter guenter  907192 2009-03-03 18:47 lib/Linux/armv5tejl/libPocoNet.so.6
-rwxr-xr-x 1 guenter guenter  293960 2009-03-03 19:11 lib/Linux/armv5tejl/libPocoNetSSL.so.6
-rwxr-xr-x 1 guenter guenter  685008 2009-03-03 18:54 lib/Linux/armv5tejl/libPocoOSP.so.2
-rwxr-xr-x 1 guenter guenter  196264 2009-03-03 18:48 lib/Linux/armv5tejl/libPocoRemoting.so.6
-rwxr-xr-x 1 guenter guenter  182932 2009-03-03 18:48 lib/Linux/armv5tejl/libPocoSoapLite.so.6
-rwxr-xr-x 1 guenter guenter  281048 2009-03-03 18:45 lib/Linux/armv5tejl/libPocoUtil.so.6
-rwxr-xr-x 1 guenter guenter  577588 2009-03-03 18:44 lib/Linux/armv5tejl/libPocoXML.so.6
-rwxr-xr-x 1 guenter guenter  353312 2009-03-03 18:54 lib/Linux/armv5tejl/libPocoZip.so.6
```

POCO Platform Benefits & Features

- Comprehensive, complete and mature C++ frameworks that save lots of work and help bringing the product to market sooner.
- Easy learning curve through intuitive, consistent and comprehensible programming interfaces, lots of sample code and good documentation.
- Native C++ code performance (no VM overhead, etc.), low memory requirements.
- Platform independence: write once – compile and run everywhere.
  - In many cases, a large part of an application (everything that does not need access to specific hardware) can be tested and debugged on the development host.
  - An application can be easily ported to a new platform.

POCO Platform/C++ vs. Java

- POCO Platform based C++ application needs significantly less resources (memory + CPU performance) than a comparable Java application (e.g, 155 MHz ARM9 with 64 MB RAM vs. 300 MHz ARM9 with 128 MB RAM) – reduced hardware cost.
- C++ applications run with native performance.
- C++ allows for easier integration with real-time code.
- C++ allows for easier integration of existing C/C++ code.
- C++ allows for easier access to specific OS features or hardware.
- No virtual machine or middleware necessary – no runtime licenses/royalties.
- Better IP protection – no decompilation of firmware (bytecode).

applied informatics

C++ Libraries, Tools and Services
to Simplify Your Life.

info@appinf.com | www.appinf.com | +43 4253 32596