

Using libpqxx:

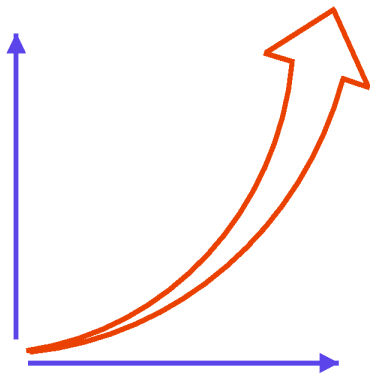
The New C++ Interface

Sunday 2003-02-09

Jeroen T. Vermeulen (jtv@xs4all.nl)

<http://www.pqxx.tk/>

<http://gborg.postgresql.org/project/libpqxx/>



No Presentation Package



42

Why I Hate My Life

- *We engaged the enemy on all sides*
- *Endless Bullet Points*
- *One dozen eggs*
- *All your base are belong to us*
- *Synergetic Customer-oriented Corporate Services-driven B2C eCommerce Solutions*
- *TM (c)*

Replacing libpq++

Idea started as abstraction layer for libpq++, in Little Brother network monitoring tool.

C++ has changed since libpq++ was written.

Small patches, discussions quickly grew into new concept.

Carte blanche to rewrite and, as it turned out, replace libpq++.

Introducing libpqxx

BSD License—same as PostgreSQL.

Runs on GNU/Linux (PowerPC, x86, SPARC),
NetBSD, MacOS X, OpenBSD, Windows. . .

Requires modern compiler: gcc 2.95 or better,
some other compilers.

The Big Idea: **Doing things the C++ way.**

Typical C++ Library Pitfalls

- “A Better C” vs. OO and Generic Programming
- Home-grown interfaces
- Access to low-level structures
 - People use underlying C API anyway
 - No elbow room in abstraction layer
- Implicitly assume detailed knowledge of underlying C API

Key Design Ideas

- Resource Acquisition Is Initialization
- No return codes to check
- Nesting instead of lifecycles

But most importantly:

- Map RDBMS concepts onto C++ concepts

Drawing a lesson from Java's popularity.

Features

Query Execution - film at eleven!

Low-level stuff - tracing, message handlers

Transactions - nontransactions, “robust” transactions

Cursors - with relative and absolute positioning

Triggers - as functor callbacks: Observer pattern

Transactors - advanced transaction management

Table streams - to read/write tables efficiently

Cached Results - for on-demand data fetching

Future Work

Large objects - coming soon

Iteration - for more classes

Non-blocking - requires new design phase

Object layer - access rows as C++ objects

Prepared statements - must remain optional

Postgres-specific - arrays, functions, stored procedures

C++ queries - generate joins etc.

On To The Code!