Lazy Application Architecture

Save Months with a
PL/SQL Framework!

The Goal

- *Encourage the Oracle community to care enough about our craft, our reputations and our free-time, to design and produce top-notch backend code, quickly and reliably, using frameworks.*

Agenda

- Laziness or Wisdom? The 80/20 Rule.
- Define application frameworks
- Database application frameworks
  - Rarity “in the wild”
  - Essential and common services
  - DIY best practices
  - Existing market
    - Intro to retail frameworks
    - Tour of the open source offerings
  - Sample App and Case Study (4X faster development, 300% better results)

- “If you want something done quickly, give it to a lazy person.” - Joe

The 80/20 rule (Pareto Principle)

- Stellar programmers (the precious “20%”) know when to build and when to re-use.
- Development of the front-end usually begins with the technical lead selecting the framework: JSF, Spring, Silverlight, etc.
  - It is a given that UI components will not be built from scratch.
- Why is the same not true of database development?
  - Familiar with QCGU, PLVision, Starter, GED?

Define: Terminology

- Framework: An application framework is a collection of software modules or components that implement common functionality used by developers to write software in a rapid, consistent manner.
- Library: A collection of related components.
- Component: A simple, robust object or routine that implements a feature of the library.

Define: Applied to DB Dev

- A component is an Oracle object: table, view, trigger, type, context, sequence, packaged routine (func/proc), etc.
- A library is typically a PL/SQL package of related routines, and the components that support that feature family.
- A framework is the entire collection of templates, standards, and libraries (packages) that offer a set of features for reliable, rapid database development.

Rarity

- Good application architecture is crucial to the aspects of deliverables that users, managers and developers care about:
  - Speed of delivery
  - Cost
  - Quality
  - Flexibility
  - Robustness
  - Scalability
  - Performance

Rarity

- This is no different from the criticality of location, site preparation, blueprints, foundation, and framing to a successful home building project.
- What happens when a structure is built without sufficient thought and investment in the site, foundation or framing?

Rarity

Rarity

- And yet, despite the values and risks, what are the first things to go when budgets and deadlines tighten?
  - Testing
  - Documentation
  - Security
  - Architecture, design and modeling
- Not what the user sees. Not an immediate problem. So these are seen as “fluff” and dispensable.

Rarity

- We are engineers, artisans and stewards.
  - Cannot allow application architecture to be cut.
- PL/SQL is mature. Let's act like it.
  - If we aren't developing with the same rigor and best practices of frontend development, it is our fault database architecture is shrugged off.
- What do frontend developers do?
  - Pair programming, regression tests, instrumentation, DRY, KISS, TDD, assertions and…
  - Re-use standard framework libraries.

Essential DB Framework Services

Needed by every database application:

- Security
- Parameters/Configuration
- Auditing
- Logging
- DBA Ops

Common DB Framework Services

Needed by those with backend processes:

- Application and Connection Metadata
- Debugging, Timing and Instrumentation
- Error Handling and Assertions
- String manipulation
- Number manipulation
- Date handling
- Messages and Email

Common Framework Services

- Locking
- IO
- Constants, Types, Cached Reference Data
- Shared SQL
- Directory Integration
- ETL
- Unit Testing
- Database code templates

Build or Buy?

- Qualities of a good framework:
  - Solid exception handling scheme
  - Good documentation, sample app, comments
  - Clean, well maintained and tested
  - Short, but intuitive library and component names
  - Layer independence and non-circular
  - Simple

The Value of Simple

Programs must be written for people to read, and only incidentally for machines to execute. - Donald Knuth

Simplicity is prerequisite for reliability. - Edsger W. Dijkstra

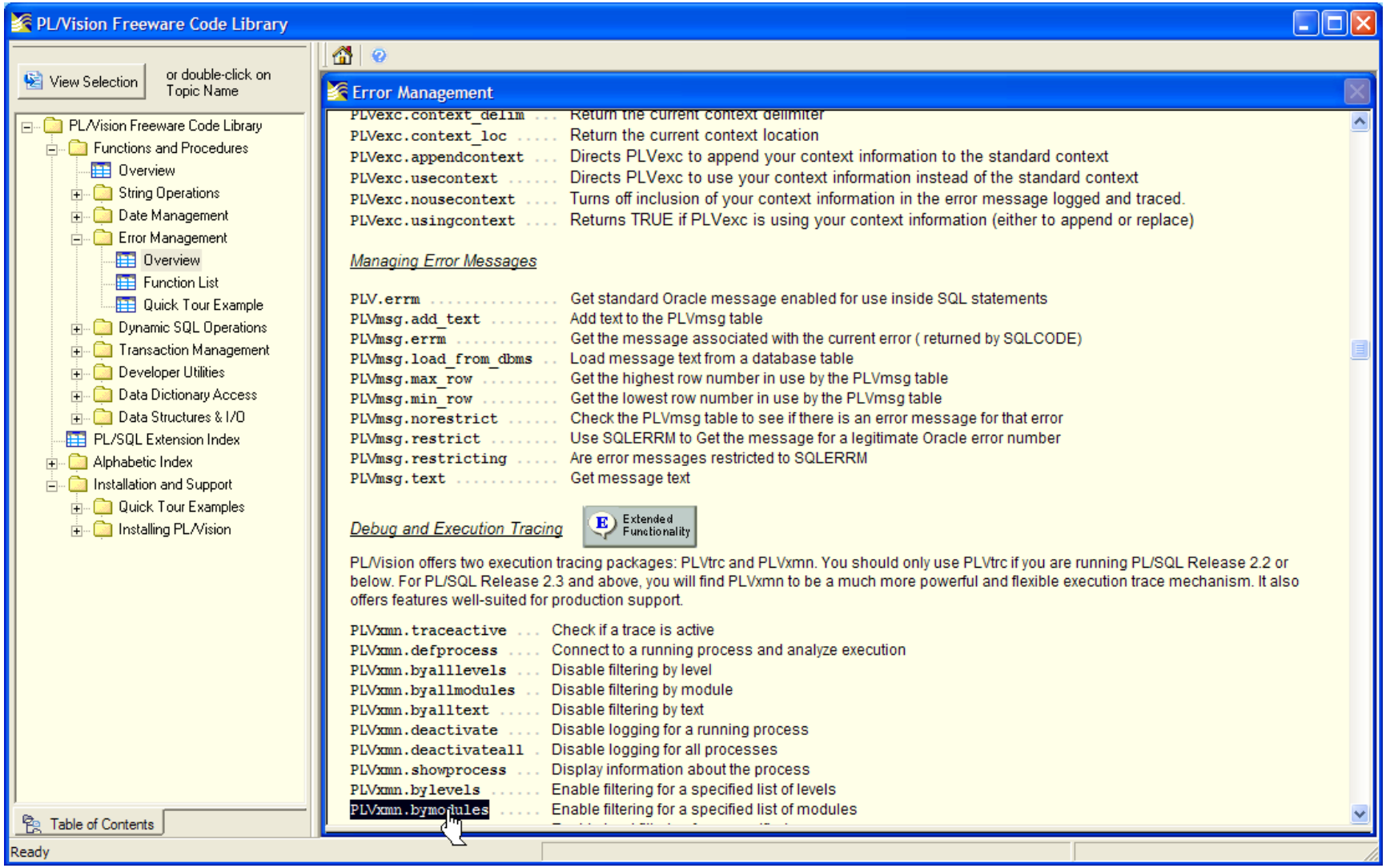
Simplicity carried to the extreme becomes elegance. - Jon Franklin

Simplicity is the ultimate sophistication. - Leonardo da Vinci

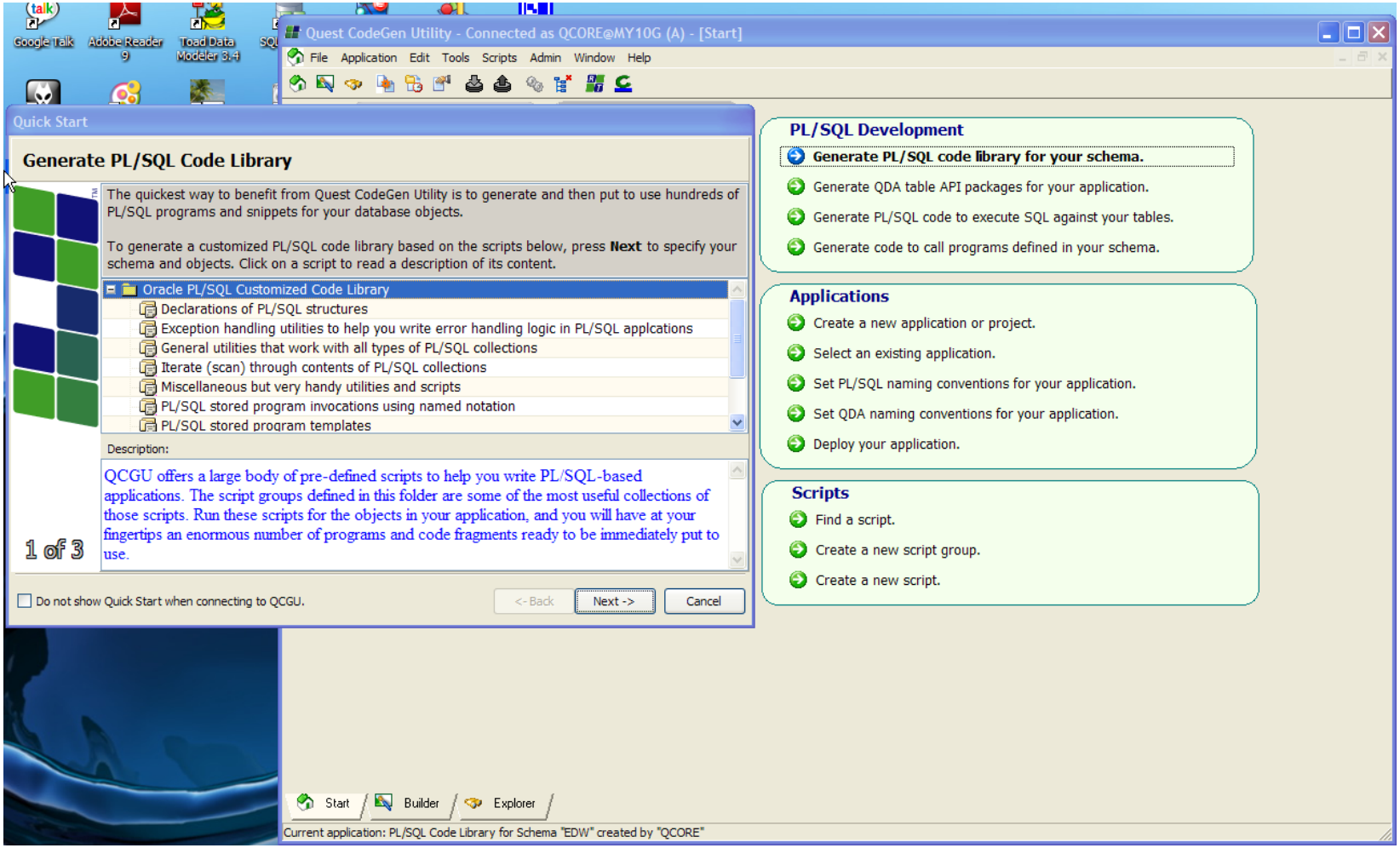
Existing Market Survey

- Retail Frameworks
  - GED Toolkit
- Free Frameworks
  - QCGU (was QNXO)
  - PLVision
- Open Source
  - PL/SQL Starter

PL/Vision

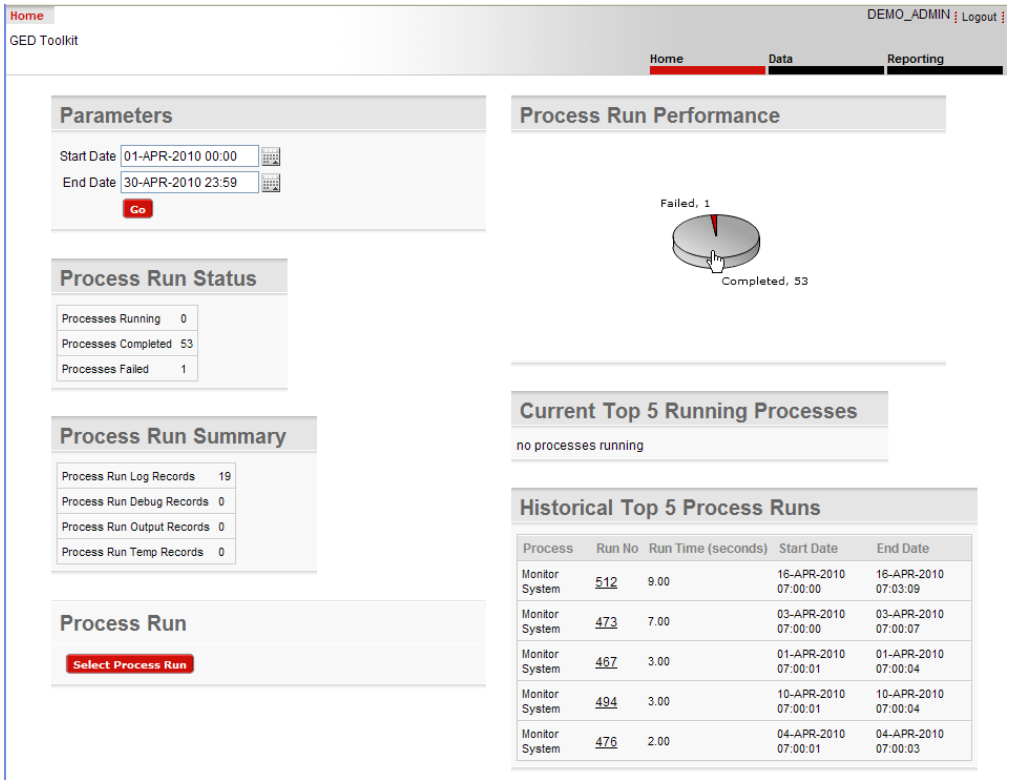

PL/Vision Freeware Code Library
View Selection
or double-click on Topic Name
PL/Vision Freeware Code Library
Functions and Procedures
Overview
String Operations
Date Management
Error Management
Overview
Function List
Quick Tour Example
Dynamic SQL Operations
Transaction Management
Developer Utilities
Data Dictionary Access
Data Structures & I/O
PL/SQL Extension Index
Alphabetic Index
Installation and Support
Quick Tour Examples
Installing PL/Vision
Table of Contents
Error Management
PLVexc.context_delim ... Return the current context delimiter
PLVexc.context_loc ..... Return the current context location
PLVexc.appendcontext ... Directs PLVexc to append your context information to the standard context
PLVexc.usecontext ..... Directs PLVexc to use your context information instead of the standard context
PLVexc.nousecontext .... Turns off inclusion of your context information in the error message logged and traced.
PLVexc.usingcontext .... Returns TRUE if PLVexc is using your context information (either to append or replace)
Managing Error Messages
PLV.errm ..... Get standard Oracle message enabled for use inside SQL statements
PLVmsg.add_text ..... Add text to the PLVmsg table
PLVmsg.errm ..... Get the message associated with the current error (returned by SQLCODE)
PLVmsg.load_from_dbms .. Load message text from a database table
PLVmsg.max_row ..... Get the highest row number in use by the PLVmsg table
PLVmsg.min_row ..... Get the lowest row number in use by the PLVmsg table
PLVmsg.norestrict ..... Check the PLVmsg table to see if there is an error message for that error
PLVmsg.restrict ..... Use SQLERRM to Get the message for a legitimate Oracle error number
PLVmsg.restricting ..... Are error messages restricted to SQLERRM
PLVmsg.text ..... Get message text
Debug and Execution Tracing
Extended Functionality
PLVision offers two execution tracing packages: PLVtrc and PLVxmn. You should only use PLVtrc if you are running PL/SQL Release 2.2 or below. For PL/SQL Release 2.3 and above, you will find PLVxmn to be a much more powerful and flexible execution trace mechanism. It also offers features well-suited for production support.
PLVxmn.traceactive ... Check if a trace is active
PLVxmn.defprocess .... Connect to a running process and analyze execution
PLVxmn.byalllevels ... Disable filtering by level
PLVxmn.byallmodules .. Disable filtering by module
PLVxmn.byalltext ..... Disable filtering by text
PLVxmn.deactivate .... Disable logging for a running process
PLVxmn.deactivateall . Disable logging for all processes
PLVxmn.showprocess ... Display information about the process
PLVxmn.bylevels ..... Enable filtering for a specified list of levels
PLVxmn.bymodules ..... Enable filtering for a specified list of modules
Ready

QCGU


Google Talk
Adobe Reader 9
Toad Data Modeler 3.4
Quest CodeGen Utility - Connected as QCORE@MY10G (A) - [Start]
File Application Edit Tools Scripts Admin Window Help
Quick Start
Generate PL/SQL Code Library
The quickest way to benefit from Quest CodeGen Utility is to generate and then put to use hundreds of PL/SQL programs and snippets for your database objects.
To generate a customized PL/SQL code library based on the scripts below, press Next to specify your schema and objects. Click on a script to read a description of its content.
Oracle PL/SQL Customized Code Library
Declarations of PL/SQL structures
Exception handling utilities to help you write error handling logic in PL/SQL applcations
General utilities that work with all types of PL/SQL collections
Iterate (scan) through contents of PL/SQL collections
Miscellaneous but very handy utilities and scripts
PL/SQL stored program invocations using named notation
PL/SQL stored program templates
Description:
QCGU offers a large body of pre-defined scripts to help you write PL/SQL-based applications. The script groups defined in this folder are some of the most useful collections of those scripts. Run these scripts for the objects in your application, and you will have at your fingertips an enormous number of programs and code fragments ready to be immediately put to use.
1 of 3
Do not show Quick Start when connecting to QCGU.
<- Back
Next ->
Cancel
PL/SQL Development
Generate PL/SQL code library for your schema.
Generate QDA table API packages for your application.
Generate PL/SQL code to execute SQL against your tables.
Generate code to call programs defined in your schema.
Applications
Create a new application or project.
Select an existing application.
Set PL/SQL naming conventions for your application.
Set QDA naming conventions for your application.
Deploy your application.
Scripts
Find a script.
Create a new script group.
Create a new script.
Start
Builder
Explorer
Current application: PL/SQL Code Library for Schema "EDW" created by "QCORE"

GED Toolkit

- Re-usable libraries. Focus on insight into backend processing: inputs, outputs, state.



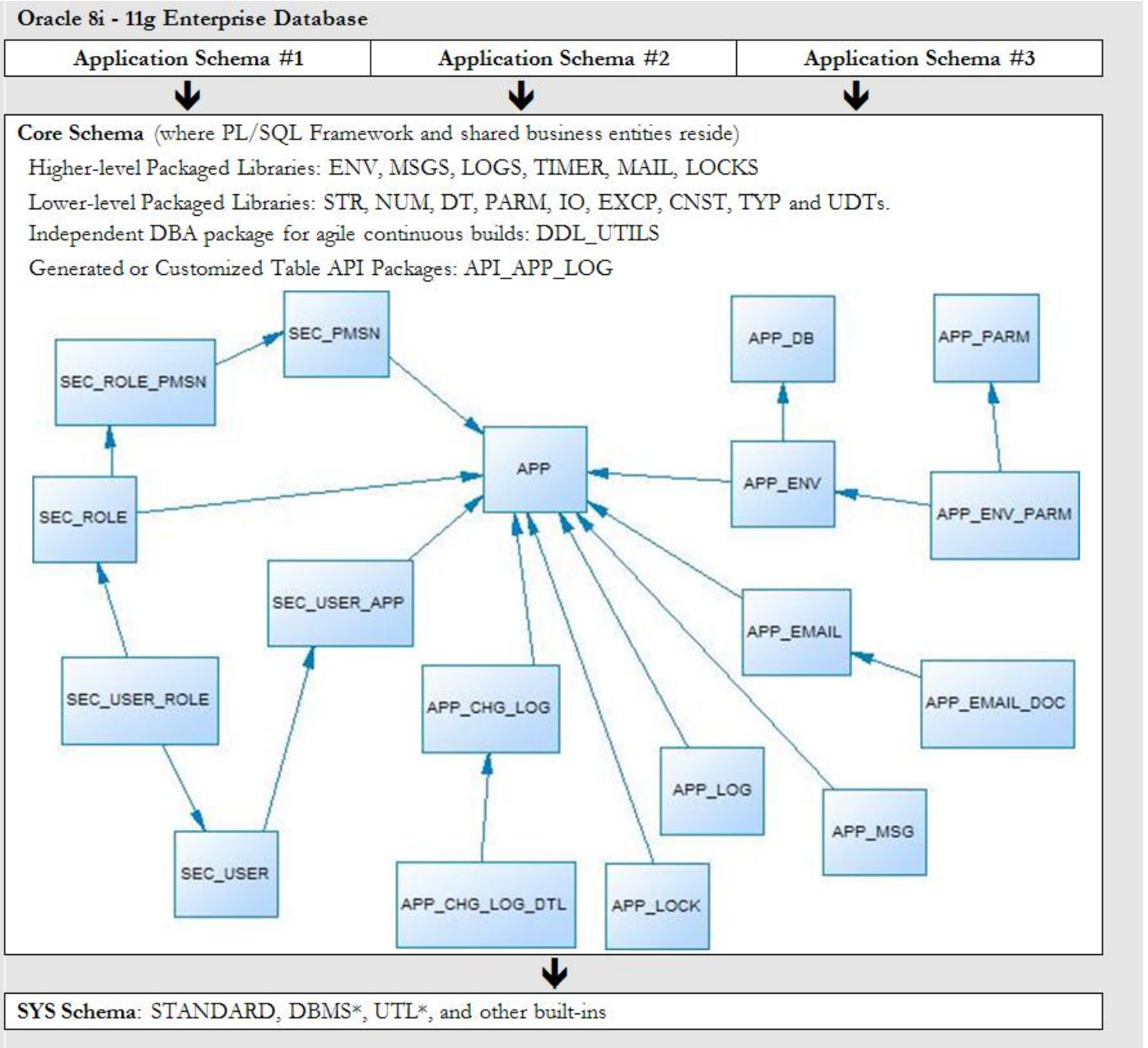
Existing Market Survey

- Retail Libraries
  - Quest Code Tester
  - PL/PDF
- Open Source or Free Libraries
  - Quest Error Manager
  - utPLSQL, PLUTO, SQL Developer Testing
  - Log4PLSQL, OraLog, Orate
  - PL/FLOW
  - PLDoc

PL/SQL "Starter"

- Author's free framework on SourceForge.
- Existed in one form or another since 1997.
- Version clock reset to v1.0 when released as open source. At 2.0 now (but 12 yrs old).
- Used in telecom, not-for-profit and energy industry.
- 2 years, 2 bugs. Simple and solid.
- Install takes 2 minutes.
- Can pick and choose some components.

Starter Schema Model



Live Tour and Case Study

- Install
- Configure
- Included sample application built on Starter:
  - Problem/Solution Repository
- Side-by-Side Case Study
  - Problem/Solution Report with File and Email Capability

Report Requirements

- Director wants new backend report that reads the problem/solution repository, writes the results to a file, and emails the results.
- File name & email subject contain today's date.
- Both should have a header with today's date.
- Report should show the problem metadata, then the solution below that.
- Report should be robust, use exception handling for IO and SMTP problems, use standardized error messages, and include debugging and performance capture ability.
- Email To static in Prod, dynamic in lower DBs.

Results

- See reports1.sql (without framework) and reports2.sql (with framework) in Starter's SampleApps\ProblemSolution folder.
- See the whitepaper for the list of problems with the viable solution in reports1.sql
- Developing with framework met 100% of the requirements, yielding 300% better code and it took 25% of the time!

Conclusion

- Frameworks essential to application architecture and can yield systems you can be proud of.
- Frameworks jumpstart new projects, saving weeks to months of risky wheel-reinventing.
  - Is that laziness? Or wisdom?
- “Starter” a decent model for features every custom Oracle application needs, but is certainly not the only framework.

Your Design Legacy

OR

Online Evaluations

- No paper this year. Hurray! ☺
- But you need internet access: ioug.org
- Session 309

Questions?

- Contact Info: Bill Coulam (bcoulam@yahoo.com)
- If interested, download PL/SQL Starter from:
  - www.dbartisans.com
  - sourceforge.net/projects/plsqlframestart
  - sourceforge.net/projects/plsqlstarter (May 2010)