



PL/SQL Application Frameworks

Presenter: *Bill Coulam*

Contact: *bcoulam@yahoo.com*

THE CHURCH OF
JESUS CHRIST
OF LATTER-DAY SAINTS



Software Architecture and Construction

- If your development people, processes, standards, tools, architecture and deliverables were buildings meant to be used by others, what would they resemble?

Software as Useful Structures



OR



OR





The Target

- My hope is to validate or renew the fire within you to produce clean, elegant, robust PL/SQL-based systems more rapidly and reliably than you could before.
- ...Or give you some ammunition needed to convince higher-ups of the same...
- So that you can enjoy your work more, or spend less time getting it done.



Agenda

- Define: Frameworks, libraries, components.
- Why the fuss over frameworks?
- Assessing new or improved frameworks.
- Framework contents
- Framework design, build and rollout
- Existing frameworks and libraries
- Case study: *PL/SQL Starter Framework*

Definitions

■ Framework

- Organized collection of libraries that implement common functionality to write software in a rapid, consistent manner.

■ Library

- Organized collection of related components.

■ Component

- Individual routine that does a single thing well.

■ Caveats

- Class libraries
- Oracle libraries



Why should I care?

- A few metaphors:

- Automobile design and construction

- Custom vs. assembly

- Home design and construction

- Custom vs. assembly vs. panel-built

- Multi-purpose facilities

- Software changes and grows far more frequently than buildings and automobiles.

Why should I care?

- Hardware, network, and database are the foundation or bedrock layer.
- Developers are the skilled builders.
- Development languages are the materials.
- The software, standards and methodology used are the power tools and blueprints.
- **The application framework is the structure's shell or "skeleton".**
- Business logic and visual interfaces are the interior and exterior treatments around the shell.

Why should I care?

- What happens when a structure is built ad-hoc, without sufficient thought and effort invested in the foundation and frame?

<non-printed slides here showing ramshackle huts and collapsed buildings>

- Why do we then build software that way?



Do we need a framework?

Yes!

...well, it depends.



Assessing need for a framework

- Timing
- Existing code base
- Existing frameworks
- Architectural direction
- Security and other policies
- Maintenance & enhancement demands
- Quality and performance

Framework Contents

■ Early stage services

- String, number, date and collection handling.
- Logging, debugging, messaging & email, exception handling, timing, assertions.
- Codes, configuration, environment.
- Locking, auditing, security
- Testing
- Standards, templates, DML API generators

■ Late stage services

- ETL-related, 3rd party integration, reporting, job management, DBA-centric utilities

■ Industry-specific services



Framework Design & Build

- Use packages to bundle related components.
- Decentralize constants and literals.
- Overloading (pseudo-polymorphism)
- Precede code with high-quality comments, testing and pseudo-code.
- Model use of your assertions, exceptions, messaging, debugging and logging library.
- Set up and model use of your templates, versioning, formatting, and testing tools.
- Follow best practices for software, especially DRY and KISS.
- Short, intuitive names for packages and routines.



Framework Design & Build

- Only catch expected exceptions.
- Use generated API packages for DML access to framework entities.
- Document framework API well, at least in the package specs if nowhere else.
- Provide small, working, sample application if you can, demonstrating all the pieces.
- Use anonymous transactions to capture critical info before rollbacks.
- Independence and loose coupling (orthogonality).



Framework Adoption

- Examine your IT environment
 - Enterprise architecture, policies and standards
 - Development methodology.
 - Management style and strategy.
 - Time, people and \$\$\$ constraints.
 - Legacy demands. New project demands.
 - 3rd party considerations.
 - Risk aversion
 - Your role, the sponsor and/or champion.

Existing Frameworks

Collection Name	License	Purpose	Location & Notes
PLNet.org	Free	Repository of PL/SQL libraries and utilities	plnet.org
PL/SQL Starter Framework	Free	My starter framework. No code-gen utilities.	sourceforge.net/projects/plsqlframestart
Feuerstein's Work	Free	Repository of new and old code generation utilities by Steven Feuerstein	stevenfeuerstein.com/gcentral.html
Quest Pipelines Archive	Free	Repository of useful PL/SQL utilities	quest-pipelines.com/pipelines/plsql/archives.htm#misc04
QCGU (Quest CodeGen Utility)	Free	Framework, Standards, Scripts, Template Factory, Code Generation, + more	qcgu.net <i>Latest incarnation of Feuerstein's vast reservoir of experience. Preceded by QXNO, PL/Vision, and PL/Generator.</i>
PL/Vision	Free	Framework, API Generator, + more	quest-pipelines.com/pipelines/dba/PLVision/plvision.htm <i>Replaced by QXNO and then QCGU. Not supported.</i>
PL/Generator	Free	Table API Generator	stevenfeuerstein.com/plgen.zip <i>Replaced by QCGU. Not supported.</i>
PL/SQL Interface Generator	Free	Table API Generator	sourceforge.net/projects/plsqlintgen

Existing Libraries

Quest CodeTester	\$\$\$	Testing Framework and Software	quest.com/code-tester-for-oracle <i>What started as Steven Feuerstein's attempt to bring agile-esque "test-first" mentality to PL/SQL (utPLSQL), has been heavily reworked, solidified, and given a UI.</i>
utPLSQL	Free	Testing framework	sourceforge.net/projects/utplsql
utPLSQL_DWH	Free	Testing framework for Data Warehousing	sourceforge.net/projects/utplsquldwh
PL/SQL Unit Test Framework	Free	Testing framework	sourceforge.net/projects/plsqlunittest
Quest Error Manager	Free	Error Handling	tinyurl.com/27xkyr <i>Included in QCGU. But offered separately as well. Not supported.</i>
Log4PLSQL	Free	Logging	log4plsql.sourceforge.net sourceforge.net/projects/log4plsql
OraLog	Free	Logging	sourceforge.net/projects/oralog
Orate	Free	Logging	sourceforge.net/projects/orate
Hotsos Library	Free	Tuning	sourceforge.net/projects/hotsos-ilo
PLDoc	Free	Documentation	sourceforge.net/projects/pldoc
PL/FLOW	Free	Workflow	sourceforge.net/projects/plflow

Existing Libraries

FTP Interface	Free	FTP from PL/SQL	myoracleportal.com/Downloads/details/id=23.html
UTL_FTP	Free	FTP from PL/SQL	sourceforge.net/projects/plsqlftp
UTL_FTP	Fair	FTP from PL/SQL	sourceforge.net/projects/utl-ftp
Mail Tools	Free	Mail from PL/SQL	myoracleportal.com/Downloads/details/id=24.html
PL/PDF	\$\$\$	PDF Generation from PL/SQL	plpdf.com
PL/Suite	\$\$\$	PL/SQL software as job schedulers, file managers and cryptographers	plsuite.com PL/Run product is unique.
Tidycode PL/SQL Formatter	\$\$\$	Code Formatter	equinoxbase.com/tpsf/
ClearSQL	\$\$\$	Code Formatter, Analyzer and Diagrammer	clearsql.com/

Case Study: Starter Framework

- First built in 1997 for 8i. Completely rebuilt for 9i in 2001. Being rebuilt again for 10g, trying to follow some of my own advice ;-)
- Open source.
 - <http://www.dbartisans.com/code.htm>
 - <https://sourceforge.net/projects/plsqlframestart>
- Be sure to fully evaluate Quest/Feuerstein's free offerings as well.
 - <http://qcggu.net>

Model of Starter Framework

