Oracle Architect Tools Project Details

Copyrights

Open Source Project Repositories

The primary repository for the project is at GitHub:

https://github.com/gaiansentience/oracle-architect-tools/

Backup repositories are at SourceForge and Bitbucket:

https://sourceforge.net/projects/ora-architect/

https://bitbucket.org/gaiansentience/oracle-architect-tools/src/master/

Contributors

Anthony Harper

Oracle Version Support

Minimum version Oracle 19c.

Supported versions: Oracle 19c and 21c.

Conditional Compilation: Any features coded using updated syntax that is incompatible with version 19 must implement conditional compilation to support the earlier version.

Project Scope and Purpose

The intent of the project is to develop a suite of PL/SQL utilities that would be useful for managing and developing PL/SQL Application Programming Interfaces within the context of Oracle Databases.

The project started with an error logging and tracing methodology that can be used to easily add instrumentation to an Oracle PL/SQL codebase.

Creating an easily deployable instrumentation utility is still a main focus of the project.

Developing code generators for common PL/SQL and SQL code is also a planned focus.

The addition of the admin utilities represents an additional focus on script generation for common architecture development and schema administration tasks.

The current approach is to deploy all of the utilities to a common schema and then making the functionality available to other schemas by granting execute rights on utility packages.

Project Areas

The project is organized into the following areas:

Database Instrumentation Systems.
Markup Language Utilities.
Code Generators.
PL/SQL Technology Samples
Administration Tools

Database Instrumentation

Error logging, program execution tracing, auditing procedure access and process timing. The basic implementation here is to use Oracle Object Types to easily add the instrumentation functionality to PL/SQL packages.

Autonomous transactions are used for all logging and instrumentation writes to the log table. This allows writing to the log table within ongoing transactions without disrupting transaction state.

Fine grained instrumentation controls using an environment settings table. Instead of using conditional compilation to toggle the use of instrumentation routines the system is designed to use an environment settings table with switches to control tracing activity. For a package that the instrumentation object has been added to, the initialization code will look up the currently active settings and use them to control writing of instrumentation data.

Markup Language Utilities

These include XML based spreadsheet generation and HTML generation using object types to format output.

Database generation of HTML is intended to map html tag generation with object types.

The initial XML prototypes have been implemented using simple string building logic.

Future development will implement the now mature Oracle XML Type methodologies.

Code Generators

This aspect of the project will focus on code generation for common architecture patterns.

Planned implementations will include:

A PL/SQL table API layer.

An XML table API exposing the table API as XML documents.

A JSON API exposing the table API as JSON documents.

PL/SQL Technology Samples

Starting with examples of Pipelined and Polymorphic Table Functions, this aspect of the project is intended to showcase a working set of examples of various methodologies available to the PL/SQL developer.

Planned implementations will include:

Pipelined Table Function examples for ETL management.

Polymorphic Table Function examples.

Examples of BULK COLLECT and BULK BIND for optimizing ETL processes.

Examples of Object Types as an approach to JSON generation or parsing.

Administration Tools

This includes a basic script building utility for partition management and miscellaneous administration tasks.

Also planned is a simplified API for deploying and managing Advanced Queuing as an approach for parallel processing ETL processes.