*Clear*DB for Oracle

*Clear*DB at a Glance

*Clear*DB automatically produces database documentation directly from an ORACLE database instance. It provides a very easy way to document all Oracle databases that a user manages, and to create a new version of the documentation whenever any database changes are made. *Clear*DB easily and quickly documents a complete set of 47 schema (tables, views, procedures, functions, packages, etc.) and non-schema (contexts, directories, profiles, users, roles, etc.) object types as well as source code modules (packaged and stand-alone subroutines) for Oracle up to 11gR2.

*Clear*DB Illustrates

ClearDB performes analyzing, formatting and illustration of PL/SQL code and includes it into the Documentation. No changes are made to the original files in the database

*Clear*DB for ORACLE is one of the most powerful Oracle database documenter known to date, which provides PL/SQL code analyzing, formatting and visualizing (Flowcharts and Call Tree diagrams and CRUD1 and CRUD2 type matrices) while at the same time creates database documentation. It helps to keep PL/SQL code reliable, testable and manageable. Industry accepted complexity measures (McCabe, Halstead and Maintainability Index, MI) are used. The code review module catches over 60 common coding mistakes and gives suggestions for good programming practices. Up to 99% of SQL and PL/SQL documented keywords and reserved words are supported by our own PL/SQL Parser. The primary output format of the database documentation is HTML. In addition to HTML, *Clear*DB Documenter produces documentation in CHM (Microsoft's Compiled HTML Help) format as well.

www.mycleardb.com

*Clear*DB and its Observation Report

ClearDB documents database anomalies and parser errors as well

The **„Database Observations“** section of the documentation represents the database objects where the Parser detected errors or alerts; or where the database status is invalid or disabled; or where the source code is wrapped.
„Parser: Objects with Errors“ displays a summary table of all invalid database objects on the selected level (database/schema/object type) and a list of detailed information about the error position and error text for each object.
„Parser: Objects with Alerts“ displays a summary table of all database objects in which the Parser found alerts on the selected level (database/schema/object type) and a list of detailed information about the warning line and warning text for each object. This report helps to quickly identify objects that require user attention.

Supports 47 Oracle DB Object Types

*Clear*DB documents a complete set of 47 schema and non-schema object types for Oracle up to version 11gR2:

CLUSTER	POLICY	FLASHBACK ARCHIVE
CONTEXT	POLICY GROUP	MATERIALIZED VIEW
DATABASE LINK	PROCEDURE	MATERIALIZED VIEW LOG
DIMENSION	PROFILE	REFRESH GROUP
DIRECTORY	QUEUE	RESOURCE CONSUMER GROUP
EDITION	ROLE	RESOURCE PLAN
FUNCTION	SEQUENCE	RESTORE POINT
INDEXTYPE	SYNONYM	ROLLBACK SEGMENT
JAVA CLASS	TABLE	SCHEDULER CHAIN
JAVA SOURCE	TABLESPACE	SCHEDULER CREDENTIAL
JOB	TRIGGER	SCHEDULER JOB
LIBRARY	TYPE	SCHEDULER JOB CLASS
OPERATOR	TYPE BODY	SCHEDULER PROGRAM
OUTLINE	USER	SCHEDULER SCHEDULE
PACKAGE	VIEW	SCHEDULER WINDOW
PACKAGE BODY		SCHEDULER WINDOW GROUP

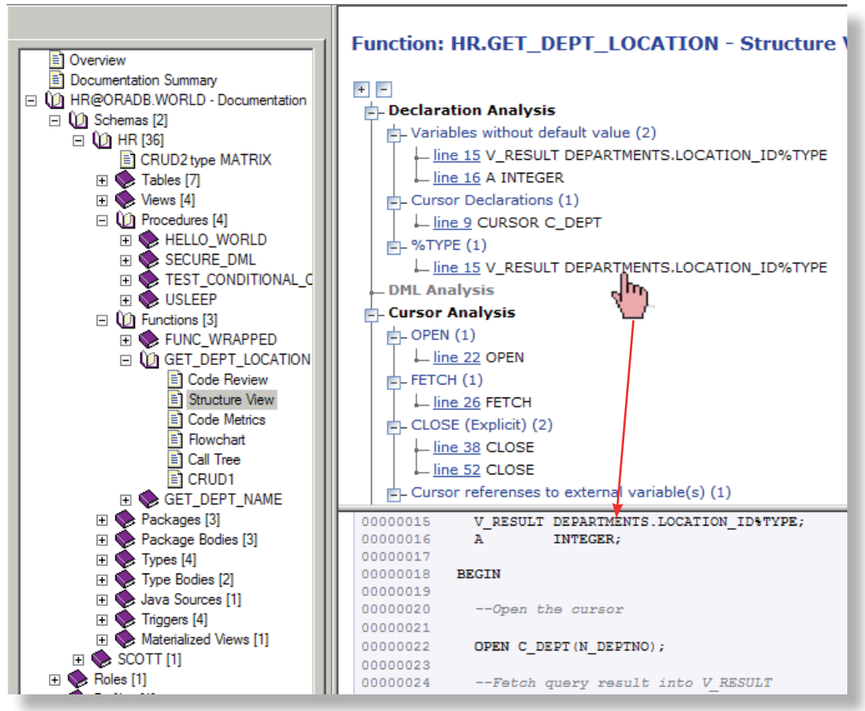
Additionally, the user may include to documentation PUBLIC SYNONYMs, PUBLIC DATABASE LINKs and RECYCLE BIN objects.

*Clear*DB and its Distinctive Features

- Generates complete and detailed database documentation for any Oracle database; supports Oracle 7 --> Oracle 11gR2
- Support for a complete set of 47 schema and non-schema object types for Oracle version up to 11gR2
- Analyzes and documents PL/SQL objects, their statuses and dependencies
- Automated code review reports parser observations (syntax errors, PL/SQL warnings, and more...)
- PL/SQL Parser (Analyzer) supports up to 99% of SQL and PL/SQL keywords and reserved words
- Includes PL/SQL code diagrams (Flowchart, Call Tree, CRUD1, CRDU2) into the database documentation
- Includes Code Metrics information of PL/SQL code
- Includes a „Code Audit“ page for each stored object
- Includes a database observation report with Parser and Database errors, alerts and warnings
- Includes a complex report about Database errors or Database anomalies
- Generates documentation in HTML (default format) and CHM formats
- Sends one or a selection of documentation from History by Email (zipped)
- Provides with an easy-to-use graphical interface that is based on step-by-step wizard technology
- Online Support Desk dramatically simplifies communication with Technical Support and keeps the application up-to-date
- Logs on to a database by using TNS, LDAP and Direct connections

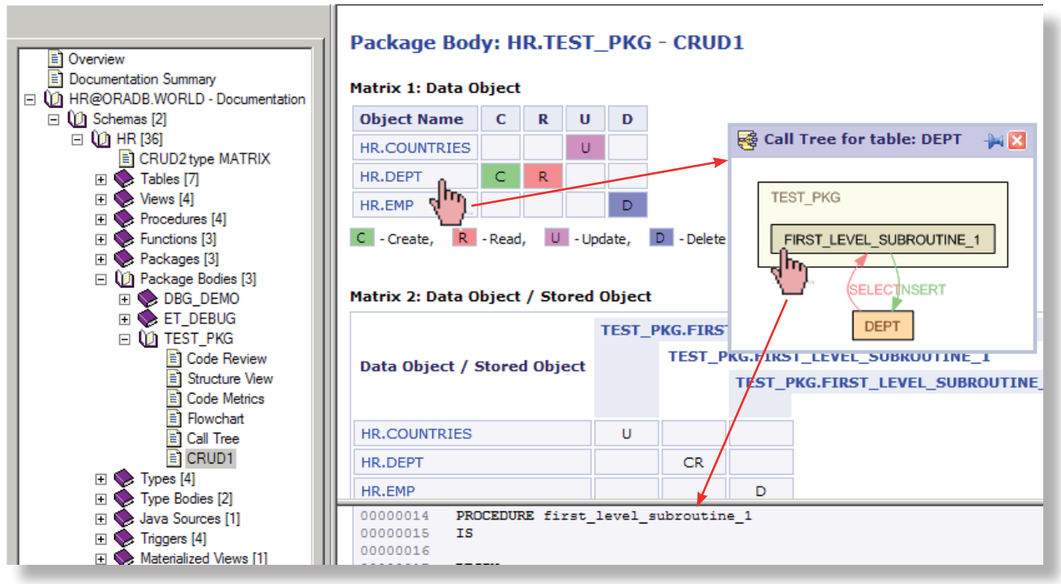
*Clear*DB Sample 1

The Code Audit Section
- *Code Review*
- *Structure View*
- *Code Metrics*
- *Flowchart*
- *Call Tree*
- *CRUD1 matrix*

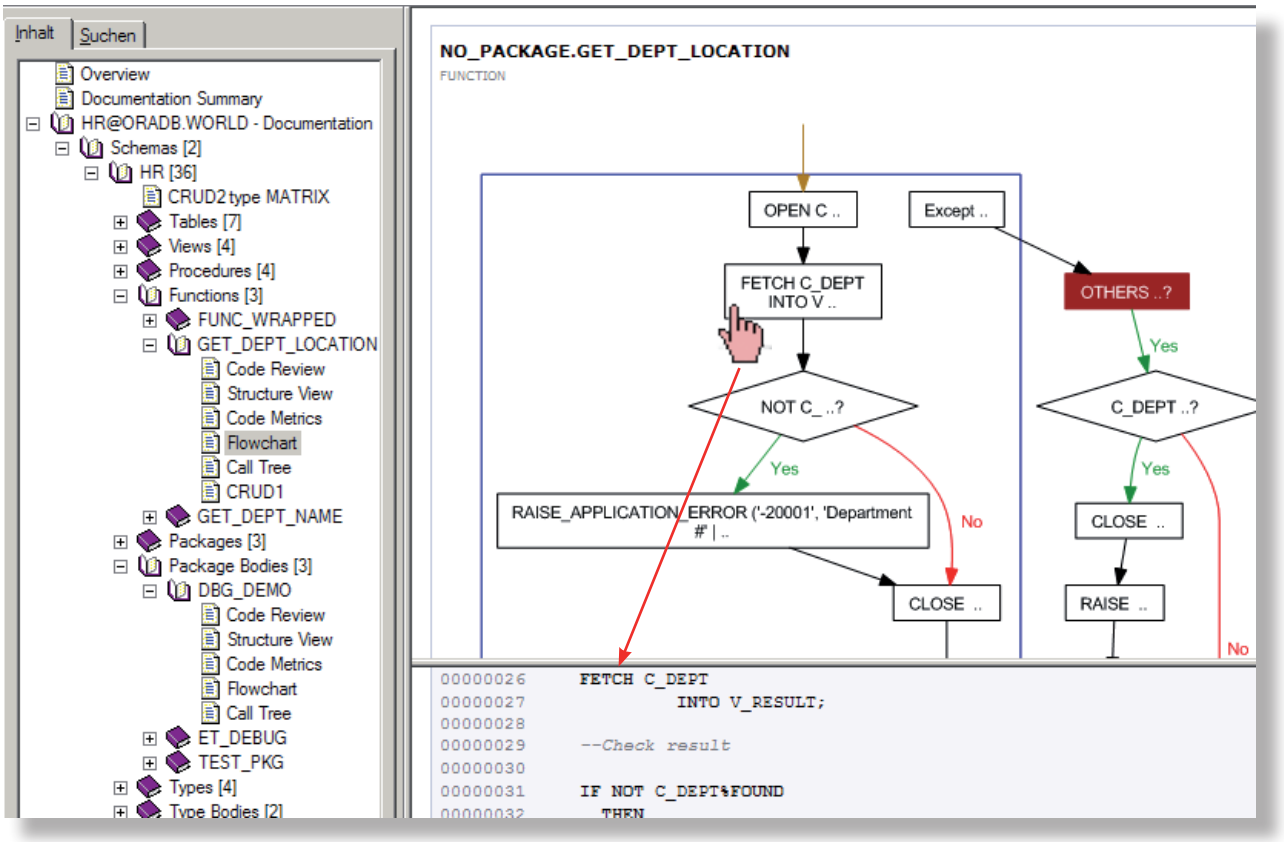


*Clear*DB Sample 2

CRUD matrix and Call Tree (clickable)



*Clear*DB Sample 3



"Click-able" PL/SQL Flowcharts, Call Tree Diagrams and CRUD Matrices

*Clear*DB uses *Clear*SQL's engine to create and include PL/SQL code Flowcharts and Call Tree diagrams that are "click-able". Click on any element within a flowchart or diagram, and the relevant line of PL/SQL code will be shown in the source code section below the flowchart. This is a superior aid for PL/SQL developers for understanding and maintaining legacy PL/SQL code. The "clickable PL/SQL diagram" feature boost's *Clear*DB's outstanding and unique position in its targeted market.

Flowchart PL/SQL

*Clear*SQL, as a plug-in in *Clear*DB, takes a package or a stand-alone subroutine and makes a set of flow-chart diagrams that visualize the code execution path. Such diagrams show the conditional branches, loops and jumps, thereby helping to understand the opaque logic. The visual patterns help to find the points of possible code refactor-ing or module restructuring, and make the reasons for high values of Cyclo-matic Complexity metrics obvious.

Call Tree Diagrams

In addition to PL/SQL flowcharts, *Clear*DB also includes Call Trees dia-grams of PL/SQL based objects in the database documentation it generates. A Call Tree diagram is a perfect aid to reading and understanding the data and control flow of legacy code.

CRUD Matrix

It documents the data elements that objects access in a database

During generation CRUD (*Create, Read, Update, Delete*) matrices are created. There are two types of CRUD matrices available in *Clear*DB: CRUD1 (on script level) and CRUD2 on (project level). CRUD matrices are clickable. Clicking on a object name in a matrix, loads the code and the relevant line of PL/SQL is highlighted in the code editor.

www.mycleardb.com

*Clear*DB - FAQ

Who is a Typical User of *Clear*DB?

- Oracle DBAs, Oracle Developers, Oracle Consultants or Project Managers who need a professional and complete set of documentation of their Oracle database. Documentation includes database structure, schemas and objects as well as their statuses and dependencies.

What Makes *Clear*DB Unique in the Market?

- In addition to the usual content about databases contained in documentation, such as database structure-schema-object descriptions, *Clear*DB is a unique Oracle Database Documenter that also includes:
  a) PL/SQL code Flowcharts (clickable, synchronized with the code)
  b) PL/SQL code Call Tree diagrams (clickable, synchronized with the code)
  c) PL/SQL code CRUDE matrices. There are two types of "clickable" CRUD matrices available in *Clear*DB:
     - CRUD1 on script level
       and
     - CRUD2 on database level
  d) Complex reports of its parser observations; e.g. invalid objects, syntax errors, PL/SQL code review warnings, and more …
  e) This parser observation report is hierarchically structured (database, schema and object type level)
- Database documentation generated by *Clear*DB offers a very easy way to quickly navigate through it. Related information is linked together and found with a single click. No need to browse through all pages.
- Users can jump directly from/to a referenced table, from error information to the specific object description/details, and much more…
- *Clear*DB is using *Clear*SQL's engine to provide a unique way of creating and maintaining Oracle database documentation on a professional level.

www.mycleardb.com

How is *Clear*DB Positioned?

- As a unique, advanced and cost-effective Oracle Database Documenter
- *Clear*DB will be an important and cost saving tool for many Oracle users
- *Clear*DB combined with *Clear*SQL offer a new unique solution for Oracle DBAs and Oracle Developers and set a new standard to keep legacy PL/SQL code working and free of errors in a very cost effective way.

software solutions
CONQUEST