

ODTUG
Kscope19 
SEATTLE, WASHINGTON • JUNE 23-27

PLEASE FILL OUT
YOUR EVALUATIONS

SEATTLE



Washington State
Convention Center

Testing with utPLSQL

Made Easy with SQL Developer

Philipp Salvisberg

Blog



@phsalvisberg



<https://www.salvis.com/blog>

BASEL • BERN • BRUGG • BUCHAREST • DÜSSELDORF • FRANKFURT A.M. • FREIBURG I.B.R. • GENEVA
HAMBURG • COPENHAGEN • LAUSANNE • MANNHEIM • MUNICH • STUTTGART • VIENNA • ZÜRICH

trivadis

■ About Me

- Trivadian since April 2000
 - Senior Principal Consultant, Partner
 - Member of the Board of Directors
 - [@phsalvisberg](https://www.salvis.com/blog)
 - <https://www.salvis.com/blog>
 - <https://github.com/PhilippSalvisberg>
- Database centric development
- Model Driven Software Development
- Author of free SQL Developer Extensions PL/SQL Unwrapper, PL/SQL Cop, utPLSQL, plscope-utils, oddgen and Bitemp Remodeler

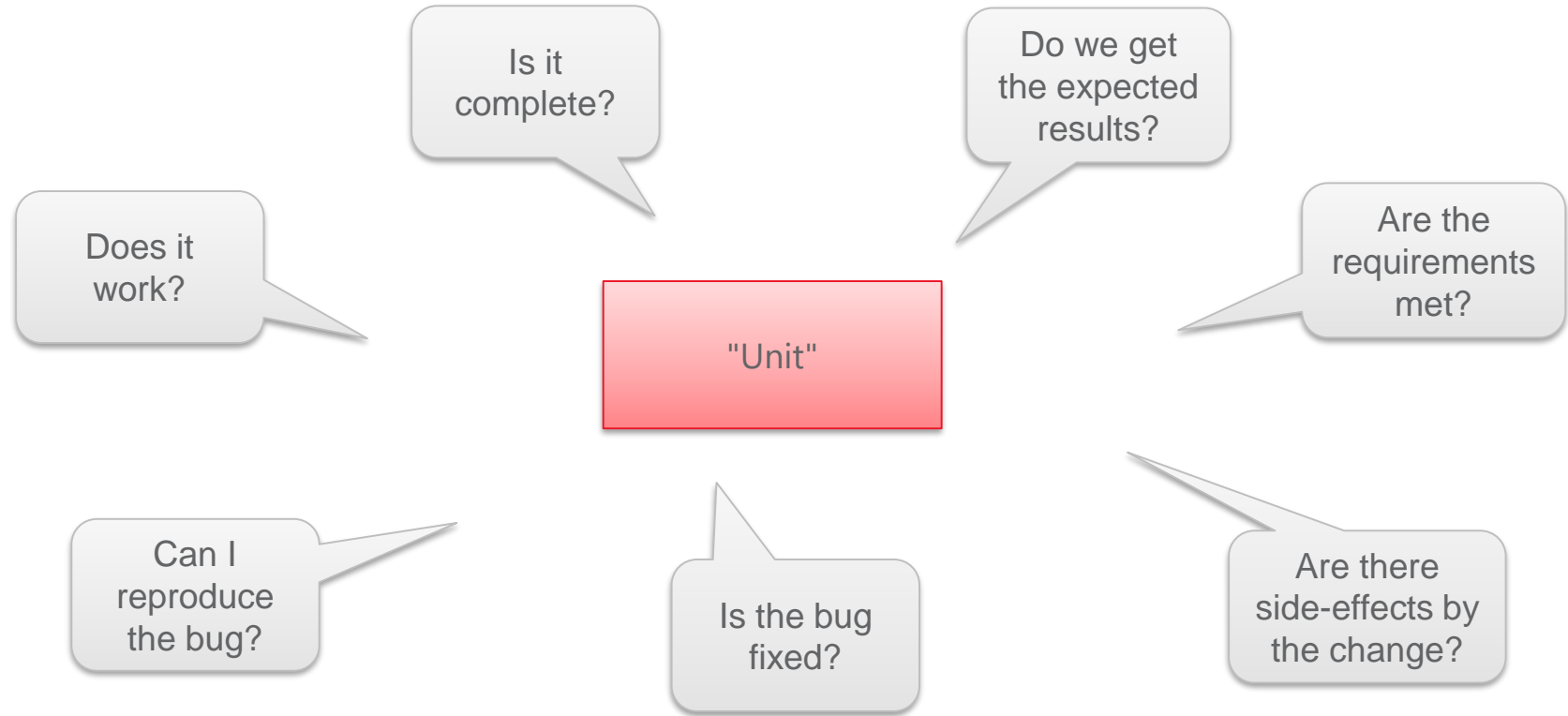


■ Agenda

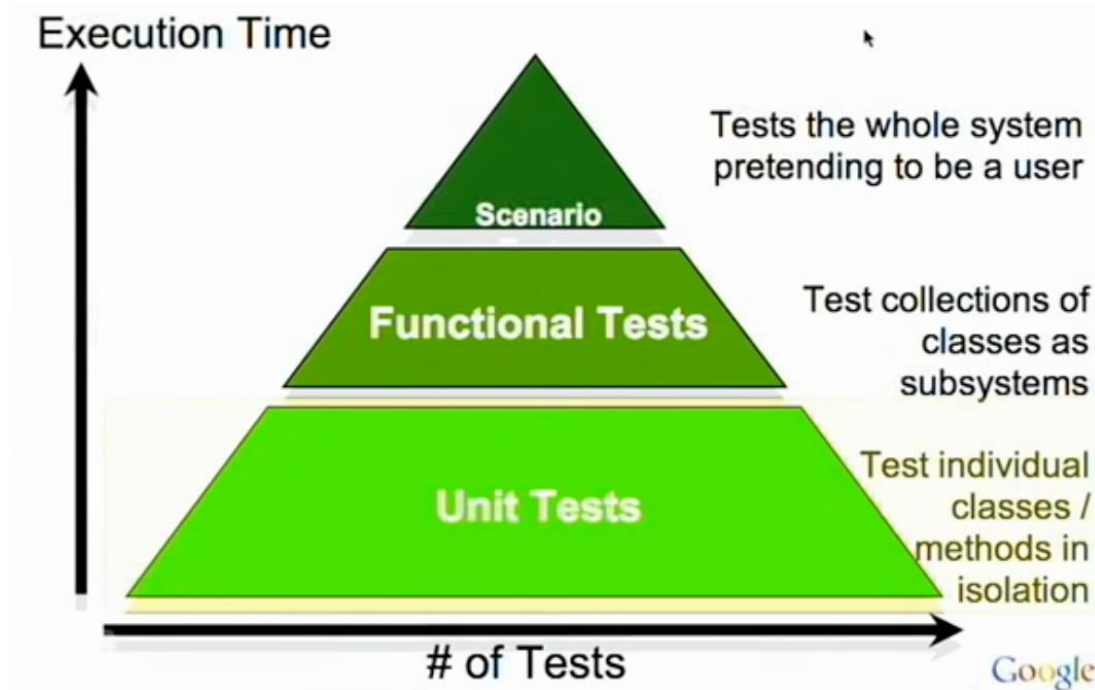
1. Introduction
2. Installation
3. Build & Run Tests in SQL Developer
4. Run Code Coverage Reports in SQL Developer
5. Realtime Reporter
6. Core Messages

Introduction

■ Why?



■ utPLSQL Test Scope



Source: Miško Hevery, The Clean Code Talks, Unit Testing, October 30, 2008,
<https://www.youtube.com/watch?v=wEhu57pih5w&t=991>

■ GUI

■ API

■ Integration

■ Components

■ Unit

utPLSQL

■ utPLSQL Units Under Test

Primary

- Types
- Packages
- Procedures
- Functions



Secondary

- Non-PL/SQL Units
- Views
- Triggers
- Tables



■ utPLSQL Suite – Open Source – Apache 2.0 License

Mandatory

- Core Testing Framework
 - Schema installed in Oracle DB
 - No repository
 - Annotation based tests

Optional

- Command Line Client
- Maven Plugin
- SQL Developer Extension



■ Test Declaration

```
CREATE OR REPLACE PACKAGE test_package_name AS
```

```
  --%suite
```

```
  --%test
```

```
  PROCEDURE procedure_name;
```

```
END;
```

--%displayname(<description>)

--%test(<description>)

--%tags(<tag>[,...])

--%throws(<exception>[,...])

--%beforeall

--%afterall

--%beforeeach

--%aftereach

--%beforetest([...])

--%aftertest([...])

--%rollback(manual)

--%disabled

--%suite(<description>)

--%suitepath(<path>)

--%tags(<tag>[,...])

--%displayame(<description>)

--%beforeall([...])

--%afterall([...])

--%beforeeach([...])

--%aftereach([...])

--%rollback(manual)

--%disabled

--%context

--%endcontext

■ Test Implementation

```
CREATE OR REPLACE PACKAGE BODY test_package_name AS
  PROCEDURE procedure_name IS
    l_actual    INTEGER := 0;
    l_expected  INTEGER := 1;
  BEGIN
    ut.expect(l_actual).to_equal(l_expected);
  END procedure_name;
END;
```

Matcher:

be_between, be_empty, be_false,
be_greater_than, be_greater_or_equal,
be_less_or_equal, be_less_than, be_like,
be_not_null, be_null, be_true, **equal**,
have_count, match

Extended options for refcursor, object
type, JSON, nested table and varray:

- include(<items>)
- exclude(<items>)
- unordered
- join_by(<items>)

■ Test Run

```
SET SERVEROUTPUT ON SIZE UNLIMITED  
EXEC ut.run('test_package_name')
```

```
test_package_name  
  procedure_name [.003 sec] (FAILED - 1)  
  
Failures:  
  
  1) procedure_name  
     Actual: 0 (number) was expected to equal: 1 (number)  
     at "TEST_PACKAGE_NAME.PROCEDURE_NAME", line 7 ut.expect(1_actual).to_equal(1_expected);  
  
Finished in .007015 seconds  
1 tests, 1 failed, 0 errored, 0 disabled, 0 warning(s)
```

Installation

■ Install utPLSQL Core Testing Framework

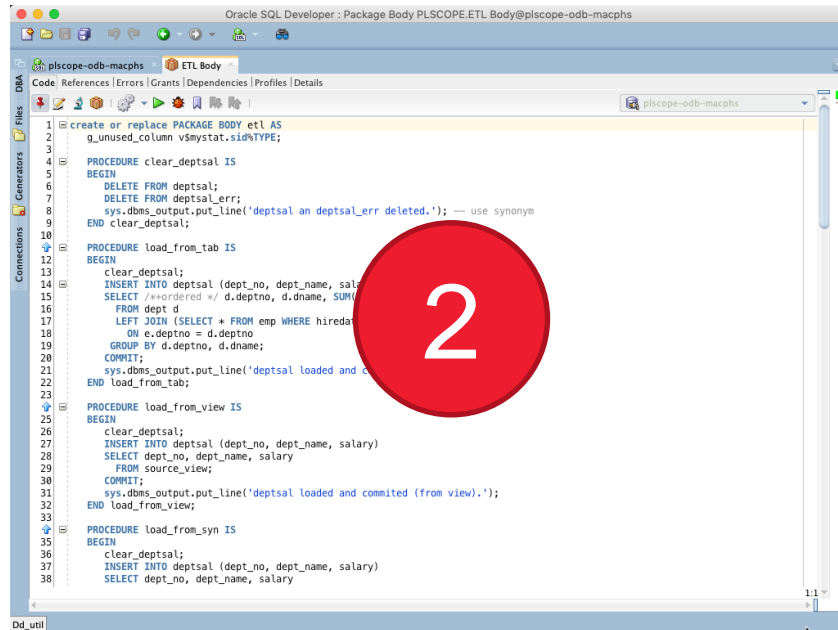
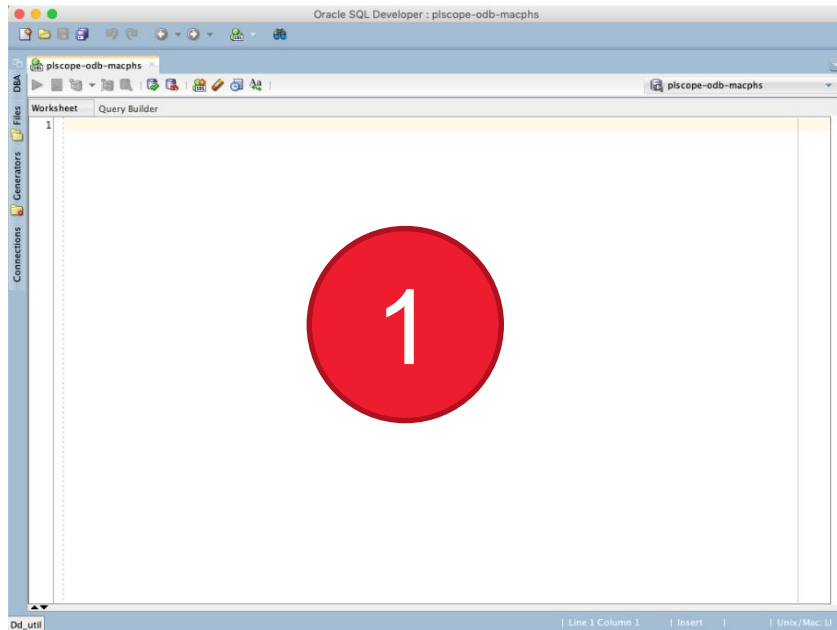
- Download utPLSQL.zip from <https://github.com/utPLSQL/utPLSQL/releases>
- Unzip utPLSQL.zip
- cd source
- sqlplus / as sysdba @install_headless.sql
 - User UT3
 - Password XNtxj8eEgA6X6b6f
 - Tablespace USERS

■ Install utPLSQL for SQL Developer

- Download utplsqli_for_SQLDev_*.zip
from <https://github.com/utPLSQL/utPLSQL-SQLDeveloper/releases>
- Start SQL Developer
- Select "Check for Updates..." in the help menu
- Use the "Install From Local File" option to install the previously downloaded "utplsqli_for_SQLDev_*.zip" file
 - User must have read/write access to SQL Developer installation directory (run as Administrator, if required)
- Restart SQL Developer

Build & Run Tests in SQL Developer

Starting Point?

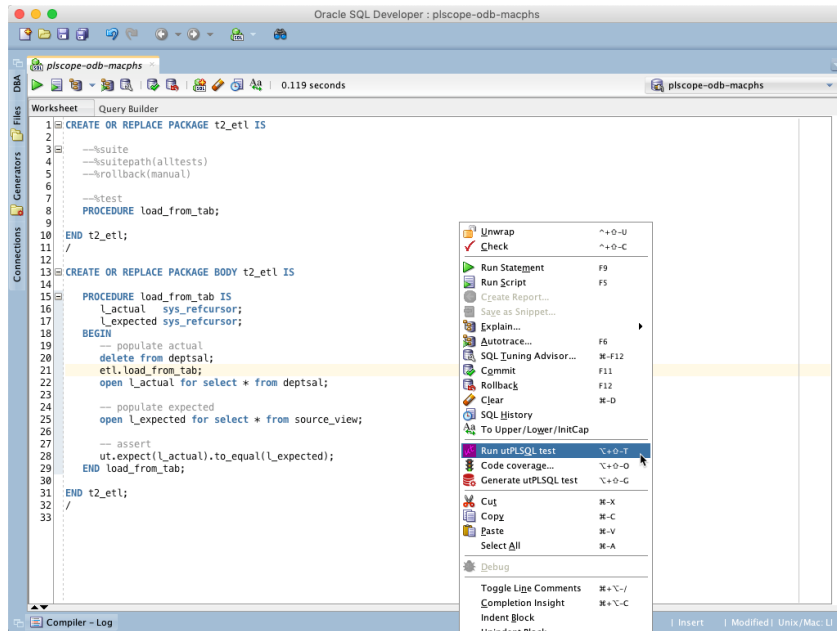


Test First – Create Test from Template

A screenshot of the Oracle SQL Developer interface. The main window displays a SQL script template for creating a test. The script starts with a comment line: `-- test [procedure_name] case 1: ... -- PROCEDURE [procedure_name]`. Below this, there's a `CREATE OR REPLACE PACKAGE BODY test_[package_name] IS` statement, followed by a `CREATE OR REPLACE PACKAGE test_[package_name] IS` statement with a `--suite` comment. A list of utility packages is shown on the left, including `SYS.utl_call_stack`, `SYS.utl_coll`, `SYS.utl_compress`, `SYS.utl_encode`, `SYS.utl_file`, `SYS.utl_gdk`, `SYS.utl_http`, `SYS.utl_i18n`, `SYS.utl_ident`, `SYS.utl_inaddr`, `SYS.utl_lms`, `SYS.utl_match`, `SYS.utl_pla`, `SYS.utl_raw`, `SYS.utl_recomp`, `SYS.utl_ref`, `SYS.utl_smtp`, `SYS.utl_sys_compress`, `SYS.utl_tcp`, `SYS.utl_url`, `SYS.utl_xml`, `UT3.ut_annotation_cache_manager`, `UT3.ut_annotation_manager`, `UT3.ut_annotation_parser`, `UT3.ut_ansiconsole_helper`, and `UT3.ut_compound_data_helper`. A tooltip is visible over the `test_[package_name]` text, showing the full package definition template.A screenshot of the Oracle SQL Developer interface showing a SQL script for creating a test. The script consists of the following lines:

```
1 CREATE OR REPLACE PACKAGE t2_package_name IS
2
3
4 --suite
5 --suitepath(alltests)
6
7 --test
8   PROCEDURE procedure_name;
9
10 END t2_package_name;
11 /
```


Test First – Complete Test & Run



Oracle SQL Developer : plscope-odb-macphs

Worksheet | Query Builder

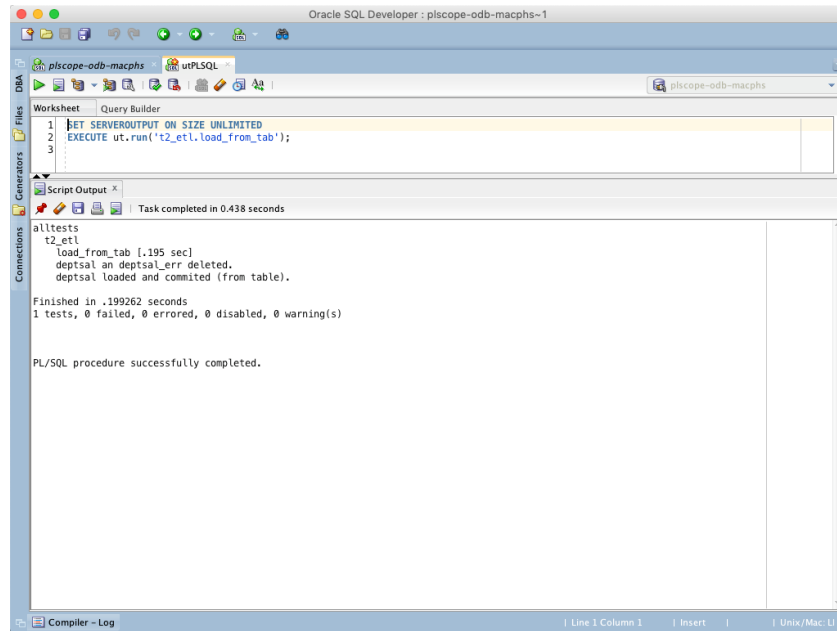
0.119 seconds

```
1 CREATE OR REPLACE PACKAGE t2_etl IS
2
3   --%suite
4   --%suitepath(alltests)
5   --%rollback(manual)
6
7   --%test
8   PROCEDURE load_from_tab;
9
10 END t2_etl;
11 /
12
13 CREATE OR REPLACE PACKAGE BODY t2_etl IS
14
15   PROCEDURE load_from_tab IS
16     l_actual sys_refcursor;
17     l_expected sys_refcursor;
18   BEGIN
19     -- populate actual
20     delete from deptsal;
21     etl.load_from_tab;
22     open l_actual for select * from deptsal;
23
24     -- populate expected
25     open l_expected for select * from source_view;
26
27     -- assert
28     ut_expect(l_actual).to_equal(l_expected);
29   END load_from_tab;
30
31 END t2_etl;
32 /
33
```

Context Menu:

- Unwrap ~+0-U
- Check ~+0-C
- Run Statement F9
- Run Script F5
- Create Report...
- Save as Snippet...
- Explain...
- Autotrace... F6
- SQL Tuning Advisor... M-F12
- Commit F11
- Rollback F12
- Clear M-D
- SQL History
- To Upper/Lower/InitCap
- Run utPLSQL test ^+0-T**
- Code coverage... ^+0-O
- Generate utPLSQL test ^+0-G
- Cut M-X
- Copy M-C
- Paste M-V
- Select All M-A
- Debug
- Toggle Line Comments M+/-
- Completion Insight M+^
- Indent Block
- Unindent Block

Compiler - Log



Oracle SQL Developer : plscope-odb-macphs-1

Worksheet | Query Builder

utPLSQL

```
1 SET SERVEROUTPUT ON SIZE UNLIMITED
2 EXECUTE ut.run('t2_etl.load_from_tab');
3
```

Script Output x

Task completed in 0.438 seconds

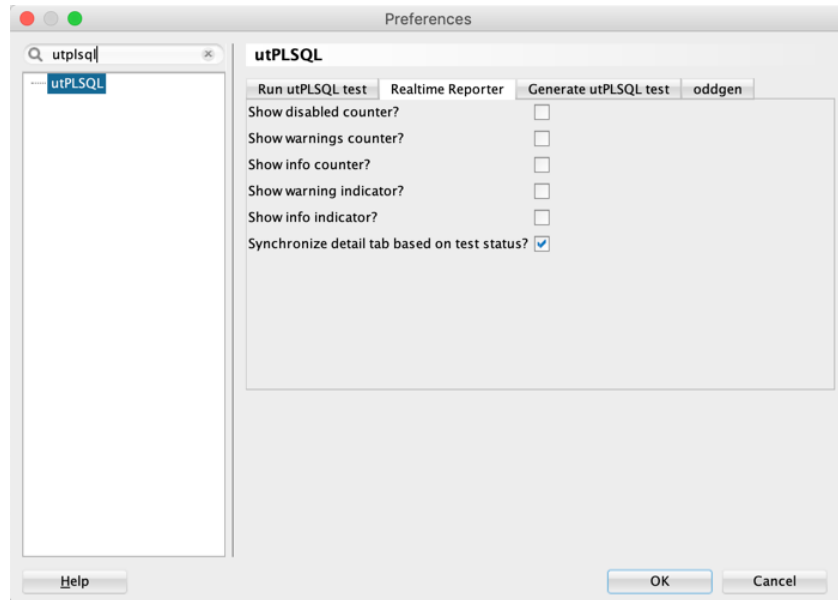
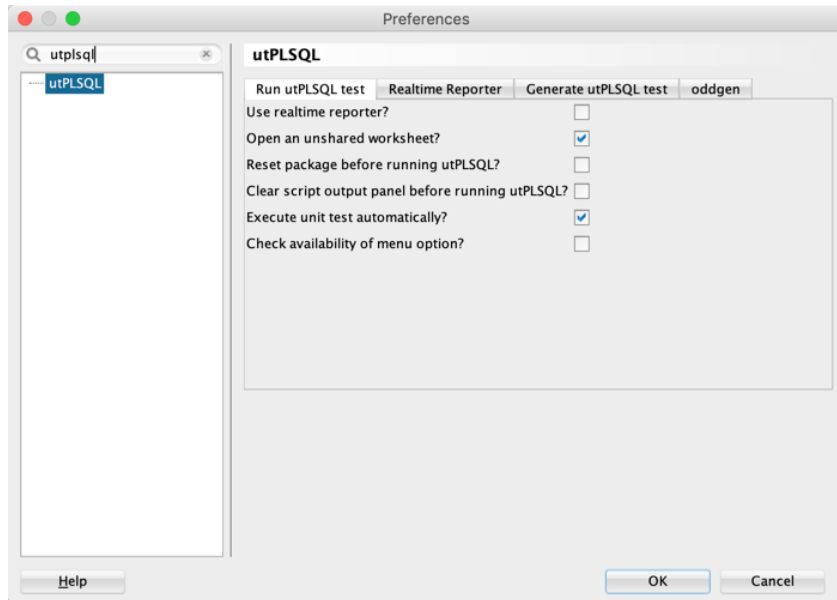
alltests
t2_etl
load_from_tab [.195 sec]
deptsal an deptsal_err deleted.
deptsal loaded and committed (from table).

Finished in .199262 seconds
1 tests, 0 failed, 0 errored, 0 warning(s)

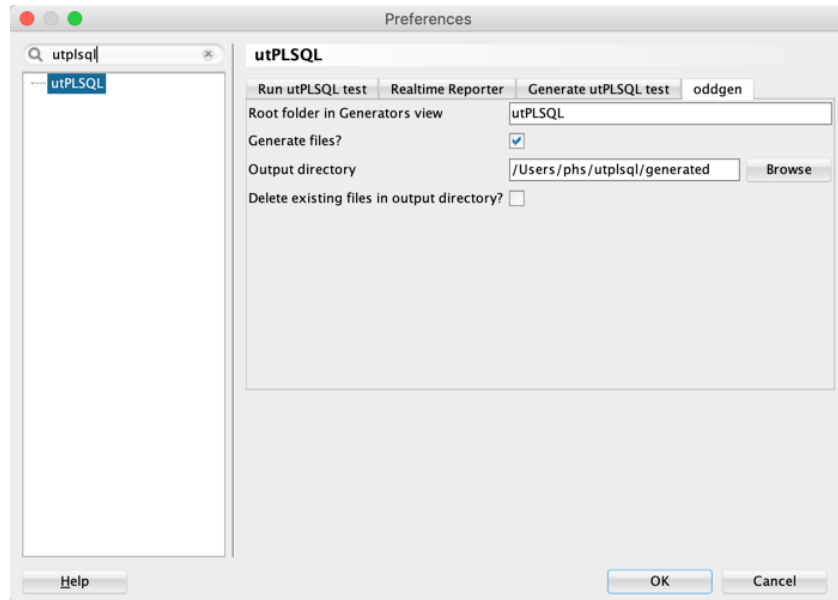
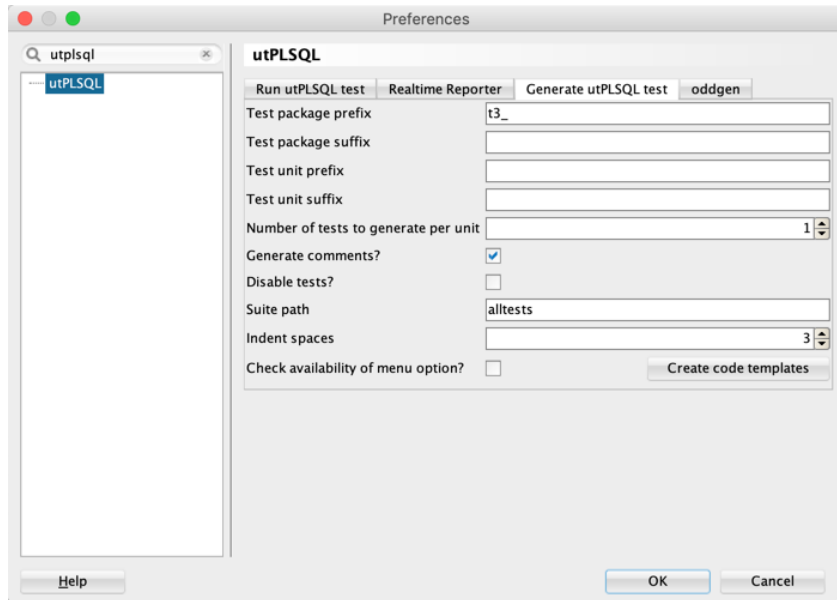
PL/SQL procedure successfully completed.

Compiler - Log

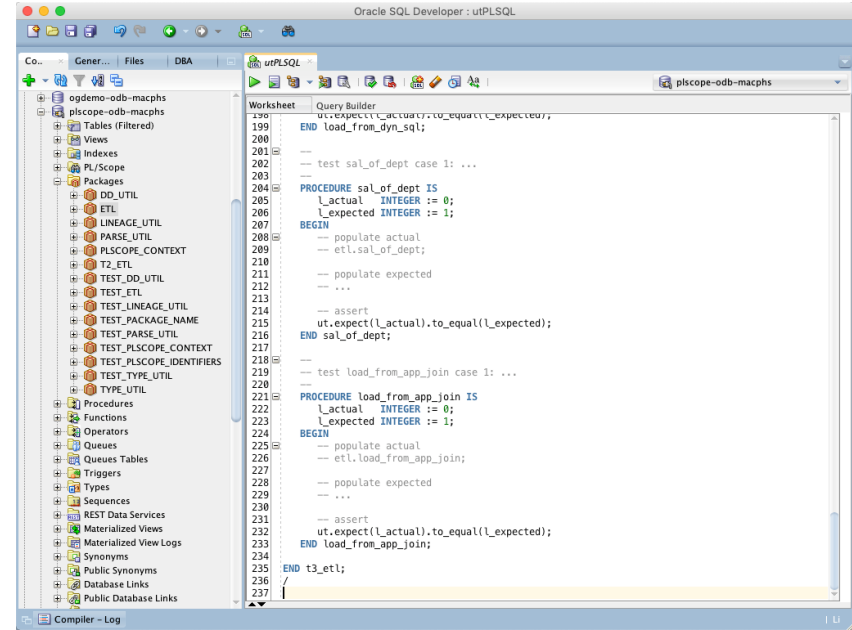
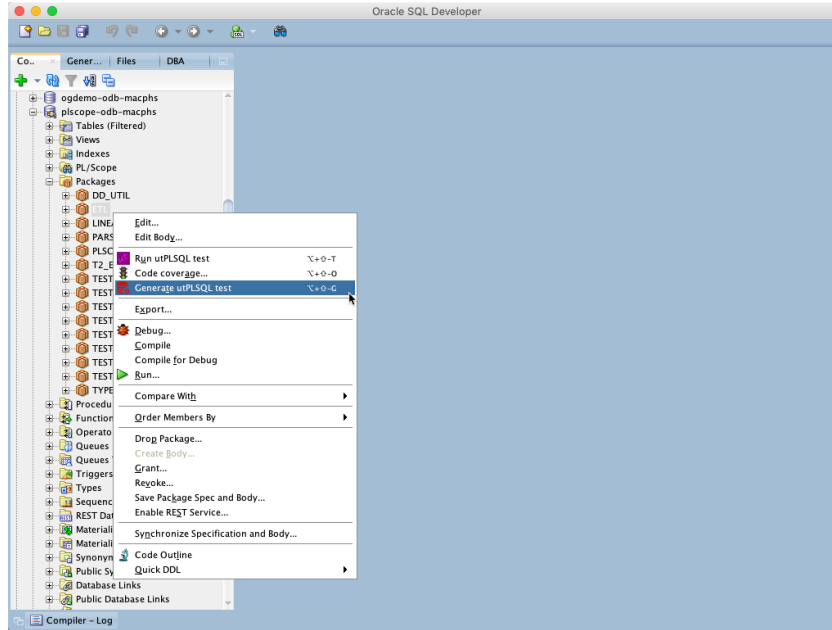
■ Configuration – Running utPLSQL Tests



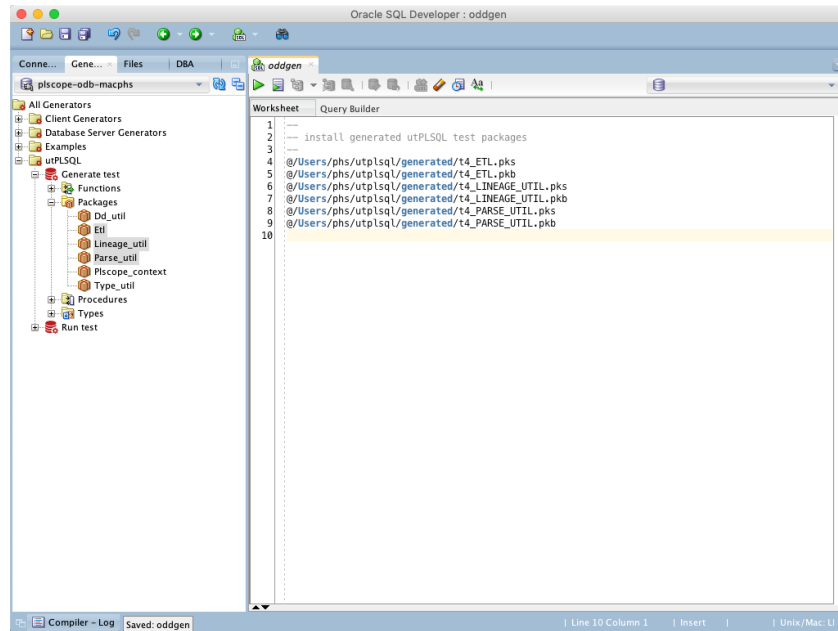
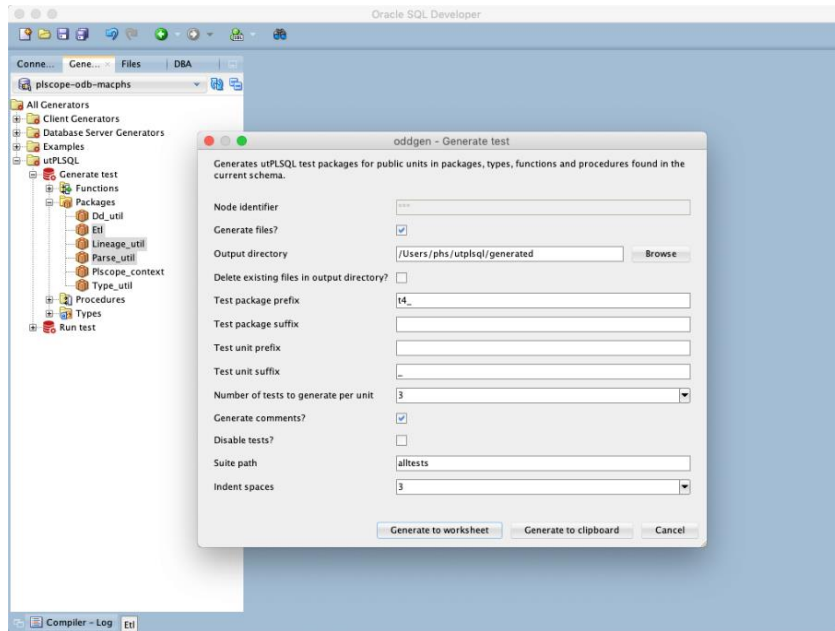
■ Configuration – Generate utPLSQL – Generating



Test Last – Create Test from Existing Code



Test Last – Generate Multiple Test Skeletons



Run Code Coverage Reports in SQL Developer

■ Code Coverage – Defintion

A measure used to describe the degree to which the source code of a program is executed when a particular test suite runs.

Source: https://en.wikipedia.org/wiki/Code_coverage

■ Line Coverage

```
CREATE OR REPLACE FUNCTION f(a IN INTEGER) RETURN INTEGER IS
BEGIN
    IF a IS NULL THEN
        RETURN 0;
    ELSE
        RETURN a*a;
    END IF;
END f;
/
```

Two test cases for
100% coverage

■ Code Block Coverage (12.2 and higher)

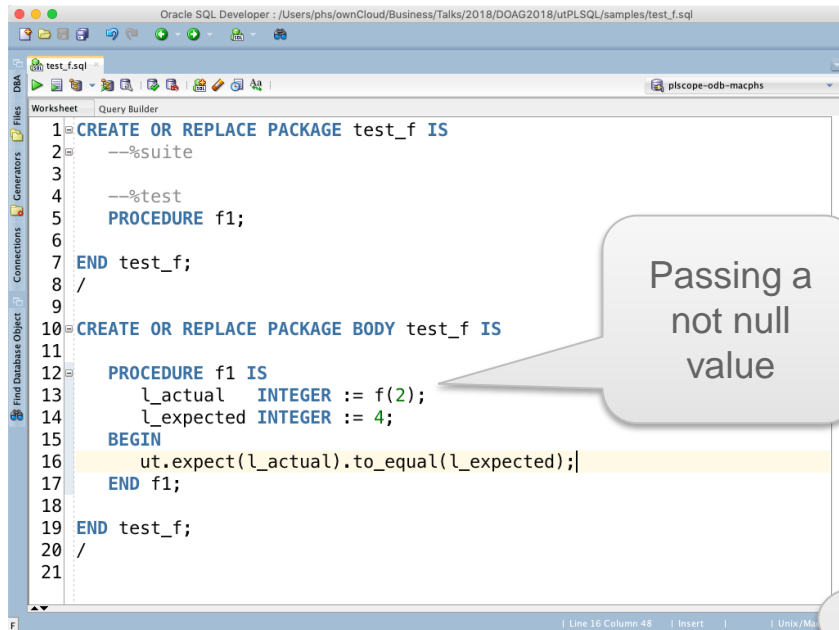
```
CREATE OR REPLACE FUNCTION f(a IN INTEGER) RETURN INTEGER IS
BEGIN
    IF a IS NULL THEN RETURN 0; ELSE RETURN a*a; END IF;
END f;
/
```

Two test cases for
100% coverage

```
CREATE OR REPLACE FUNCTION f(a IN INTEGER) RETURN INTEGER IS
BEGIN
    RETURN coalesce(a*a, 0);
END f;
/
```

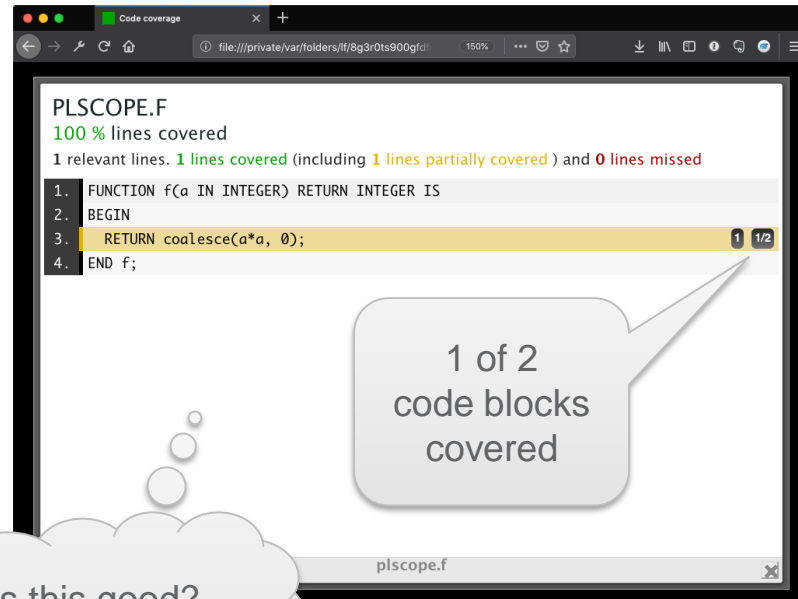
One test case for
100% coverage
when passing NULL

utPLSQL Combines Line & Code Block Coverage



```
1 CREATE OR REPLACE PACKAGE test_f IS
2   --%suite
3
4   --%test
5   PROCEDURE f1;
6
7 END test_f;
8 /
9
10 CREATE OR REPLACE PACKAGE BODY test_f IS
11
12   PROCEDURE f1 IS
13     l_actual INTEGER := f(2);
14     l_expected INTEGER := 4;
15   BEGIN
16     ut.expect(l_actual).to_equal(l_expected);
17   END f1;
18
19 END test_f;
20 /
21
```

Passing a not null value



PLSCOPE.F
100 % lines covered
1 relevant lines. 1 lines covered (including 1 lines partially covered) and 0 lines missed

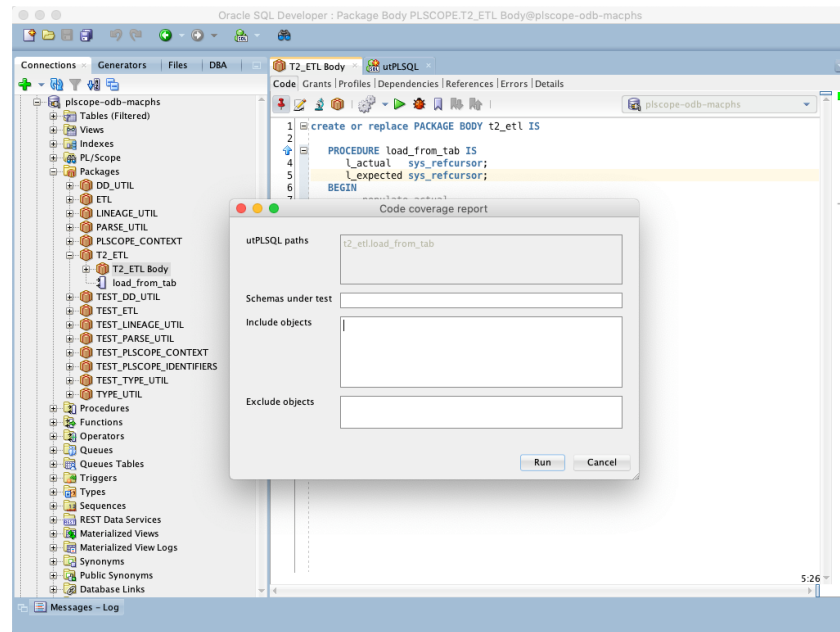
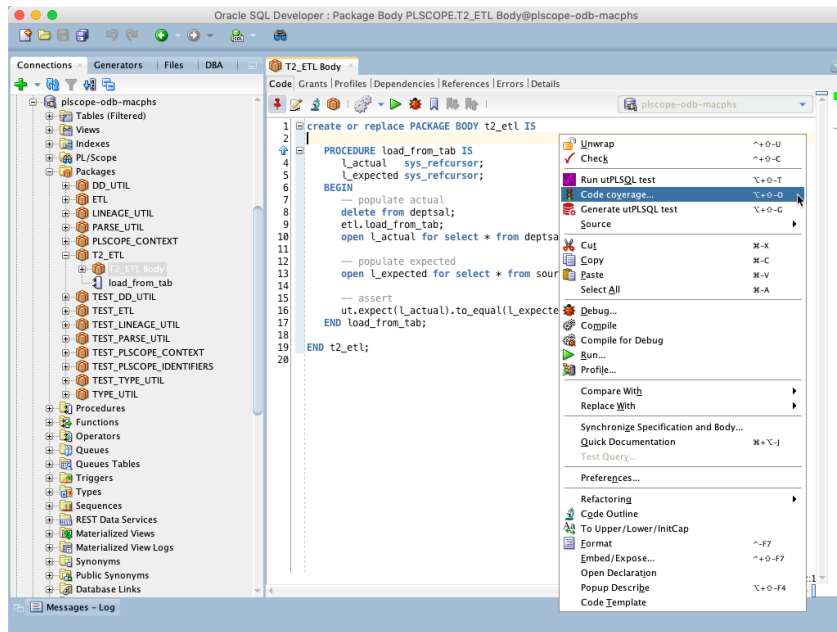
```
1. FUNCTION f(a IN INTEGER) RETURN INTEGER IS
2. BEGIN
3.   RETURN coalesce(a*a, 0);
4. END f;
```

1 of 2 code blocks covered

Is this good? helpful?



Run Code Coverage Report





Code Coverage Report

Code coverage

file:///private/var/folders/lf/8g3r0ts900gdfn2xxkn9yz0000/... Generated less than a minute ago

All files (.77%)

6 files in total.
913 relevant lines. 7 lines covered and 906 lines missed.

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
plscope.dd_util	0 %	147	147	0	147	0
plscope.lineage_util	0 %	358	358	0	358	0
plscope.parse_util	0 %	216	216	0	216	0
plscope.plscope_context	0 %	43	43	0	43	0
plscope.type_util	0 %	96	96	0	96	0
plscope.etl	13.21 %	147	53	7	46	0

Showing 1 to 6 of 6 entries

Generated by utPLSQL v3.1.3.2297-develop
Based on simplecov-htm v0.10.0

Code coverage

file:///private/var/folders/lf/8g3r0ts900gdfn2xxkn9yz0000/... PLSCOPE.ETL

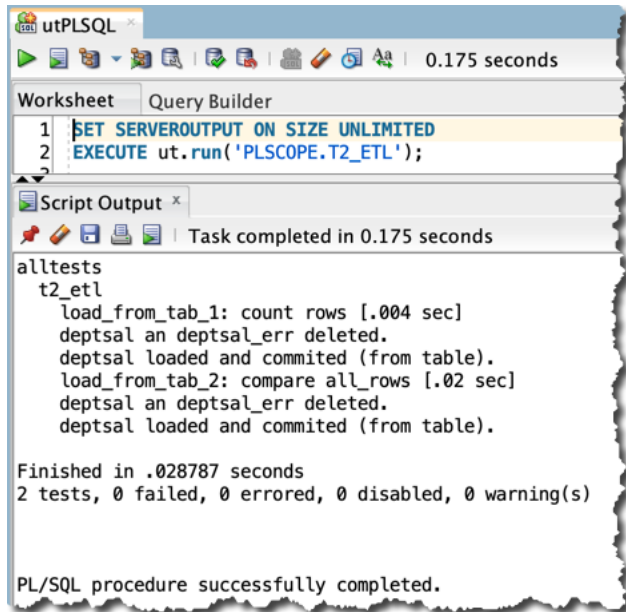
13.21 % lines covered
53 relevant lines. 7 lines covered and 46 lines missed

```
1. PACKAGE BODY etl AS
2.   g_unused_column v$mystat.sid%TYPE;
3.
4. PROCEDURE clear_deptsal IS
5. BEGIN
6.   DELETE FROM deptsal;
7.   DELETE FROM deptsal_err;
8.   sys.dbms_output.put_line('deptsal an deptsal_err deleted. '); -- use synonym
9. END clear_deptsal;
10.
11. PROCEDURE load_from_tab IS
12. BEGIN
13.   clear_deptsal;
14.   INSERT INTO deptsal (dept_no, dept_name, salary)
15.   SELECT /*ordered */ d.deptno, d.dname, SUM(e.sal + NVL(e.comm, 0)) AS sal
16.   FROM dept d
17.   LEFT JOIN (SELECT * FROM emp WHERE hiredate > DATE '1980-01-01') e
18.   ON e.deptno = d.deptno
19.   GROUP BY d.deptno, d.dname;
20.   COMMIT;
21.   sys.dbms_output.put_line('deptsal loaded and committed (from table). ');
22. END load_from_tab;
23.
```

plscope.etl

Realtime Reporter

Idea



```
1 SET SERVEROUTPUT ON SIZE UNLIMITED
2 EXECUTE ut.run('PLSCOPE.T2_ETL');
```

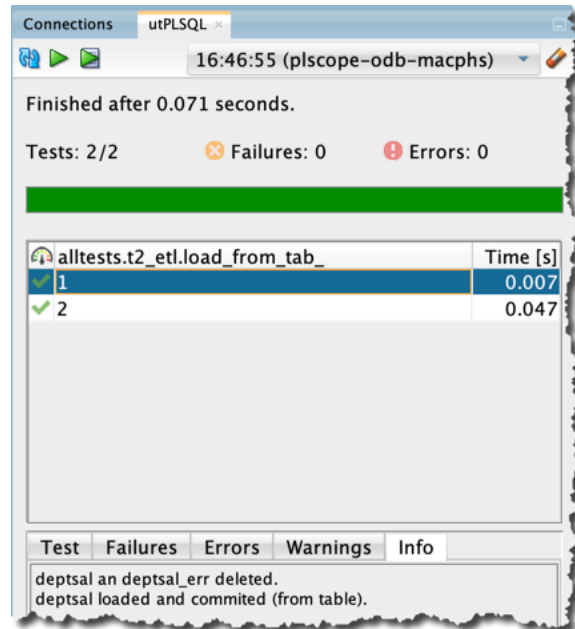
Script Output x

Task completed in 0.175 seconds

alltests
t2_etl
load_from_tab_1: count rows [.004 sec]
deptsal an deptsal_err deleted.
deptsal loaded and committed (from table).
load_from_tab_2: compare all_rows [.02 sec]
deptsal an deptsal_err deleted.
deptsal loaded and committed (from table).

Finished in .028787 seconds
2 tests, 0 failed, 0 errored, 0 disabled, 0 warning(s)

PL/SQL procedure successfully completed.



Connections utPLSQL x

16:46:55 (plscope-odb-macphs)

Finished after 0.071 seconds.

Tests: 2/2 Failures: 0 Errors: 0

	Time [s]
1	0.007
2	0.047

Test Failures Errors Warnings Info

deptsal an deptsal_err deleted.
deptsal loaded and committed (from table).

■ utPLSQL Reporters

Features

- Show tests results
 - Tests
 - Suites
 - Run summary
- Use-case driven
 - Content
 - Output format
- Multiple reports for a single run

Supplied Reporters

- Documentation reporter (Default)
- Test result output formats for JUnit, Teamcity, TFS / VSTS
- Code coverage output formats for SonarQube, Coveralls, Copertura, standalone HTML report
- Debug Reporter
- Realtime Reporter

Realtime Reporter



utPLSQL

16:26:52 (plscope-odb-macphs)

Finished after 2.000 seconds.

Tests: 95/95 Failures: 1 Errors: 0

Suitepath	Time [s]
alltests.t3_dd_util.get_column_id.get_column_id_2	0.000
alltests.t2_etl.load_from_tab_1	0.007
alltests.t2_etl.load_from_tab_2	0.056
plscope.test.test_type_util.test_dedup_t_obj	0.026
plscope.test.test_type_util.test_dedup_t_col	0.029
plscope.test.test_type_util.test_dedup_t_col_lineage	0.034

Test Failures Errors Warnings Info

Assert description (failed line)

1Line 35

Actual: refcursor [count = 4] was expected to equal: refcursor [count = 3]

Diff:

Rows: [1 differences]

Extra: <DEPT_NO>40</DEPT_NO> <DEPT_NAME>OPERATIONS</DEPT_NAME> <SALARY>

at "PLSCOPE.T2_ETL.LOAD_FROM_TAB_2", line 35 ut.expect(l_actual).to_equal(l_expected).unordered;

utPLSQL T2_ETL Body

Code References | Profiles | Details | Errors | Grants | Dependencies

plscope-odb-macphs

```
16 ut.expect(l_actual).to_equal(l_expected);
17 END load_from_tab_1;
18
19 --
20 -- test - compare rows
21
22
23 PROCEDURE load_from_tab_2 IS
24   l_actual sys_refcursor;
25   l_expected sys_refcursor;
26 BEGIN
27   -- populate actual
28   delete from deptsal;
29   etl.load_from_tab;
30   open l_actual for select * from deptsal order by dept_no;
31
32   -- populate expected
33   open l_expected for select * from source_view where dept_no != 40 order by d
34
35   -- assert
36   ut.expect(l_actual).to_equal(l_expected).unordered;
37 END load_from_tab_2;
38
39 END t2_etl;
PACKAGE BODY t2_etl PROCEDURE load_from_tab_2 BEGIN
```

35:1

Core Messages

■ The First Step Is the Hardest

- Set up a test-friendly environment
 - Install utPLSQL core testing framework
 - Install SQL Developer for utPLSQL
- Start with tests
 - to reproduce bugs
 - for new requirements



Questions and Answers...

Philipp Salvisberg
Senior Principal Consultant

Tel. +41 58 459 52 31

Philipp.salvisberg@trivadis.com



@phsalvisberg



<https://www.salvis.com/blog>

trivadis

ODTUG
Kscope19 
SEATTLE, WASHINGTON • JUNE 23-27

**PLEASE FILL OUT
YOUR EVALUATIONS**

SEATTLE

 Washington State
Convention Center