

TESTING WITH UTPLSQL MADE EASY WITH SQL DEVELOPER

Philipp Salvisberg
17th August 2021

2 WELCOME



PHILIPP SALVISBERG SENIOR PRINCIPAL CONSULTANT

- Database centric development
- Model Driven Software Development
- Author of free SQL Developer Extensions
PL/SQL Unwrapper, db* CODECOP, utPLSQL,
plscope-utils, oddgen and Bitemp Remodeler



INTRODUCTION

4 TEST AUTOMATION

"The use of software
separate from the software being tested
to **control the execution of tests**
and the comparison of
actual outcomes with predicted outcomes."

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

5 TESTING SCOPE IN DATABASE DEVELOPMENT

- Every component of an application that is deployed in the database



- Types
- Packages
- Procedures
- Functions
- Views
- Triggers
- Constraints



Core Testing Framework

- Schema in the data
- No repository
- Annotation based tests



Development

- Realtime Reporter
- Code Coverage, Code Templates, etc.



Test Automation

- Command Line Client
- Maven Plugin
- Varous Reporters



7 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

8 TEST IMPLEMENTATION

```
CREATE OR REPLACE PACKAGE BODY test_package_name AS
  PROCEDURE procedure_name IS
    l_actual    SYS_REFCURSOR;
    l_expected  SYS_REFCURSOR;
  BEGIN
    open l_actual for select * from dept where deptno != 30;
    open l_expected for select * from dept;
    ut.expect(l_actual).to_equal(l_expected).join_by('DEPTNO');
  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, contain, **equal**,
have_count, match

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

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

9 TEST RUN

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

```
test_package_name
  procedure name [.03 sec] (FAILED - 1)
```

Failures:

1) procedure name

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

Diff:

Rows: [1 differences]

```
PK <DEPTNO>30</DEPTNO> - Missing: <DEPTNO>30</DEPTNO><DNAME>SALES</DNAME><LOC>CHICAGO</LOC>  
at "SCOTT.TEST_PACKAGE_NAME.PROCEDURE_NAME", line 8 ut.expect(l_actual).to_equal(l_expected)  
                                     .join by('DEPTNO');
```

Finished in .034792 seconds

```
1 tests, 1 failed, 0 errored, 0 disabled, 0 warning(s)
```

List of ...

- [schema.]**package**[.procedure]
- [schema]:suitepath[.context][.procedure]

List of ...

- tag (include)
- -tag (exclude)

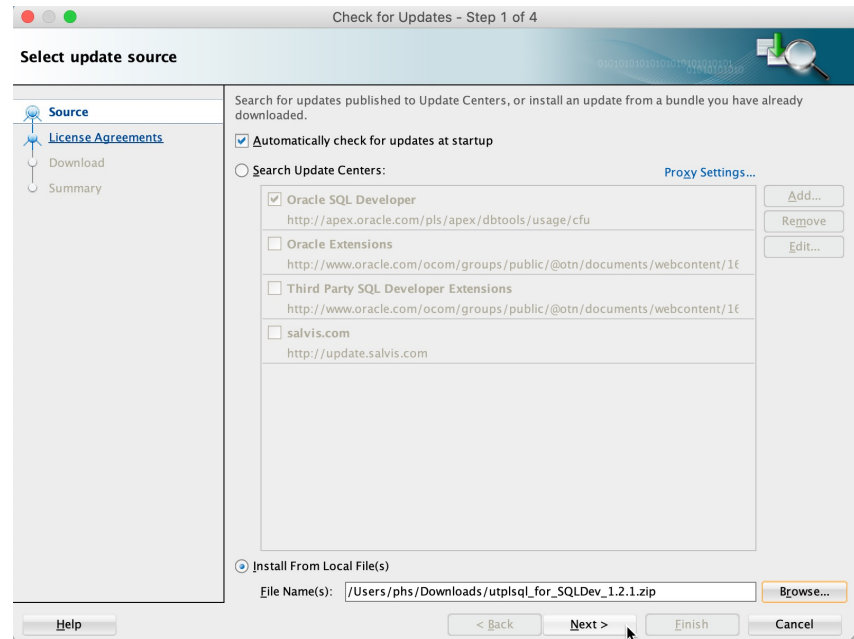
INSTALLATION

11 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

12 INSTALL utPLSQL FOR SQL DEVELOPER FROM FILE *)

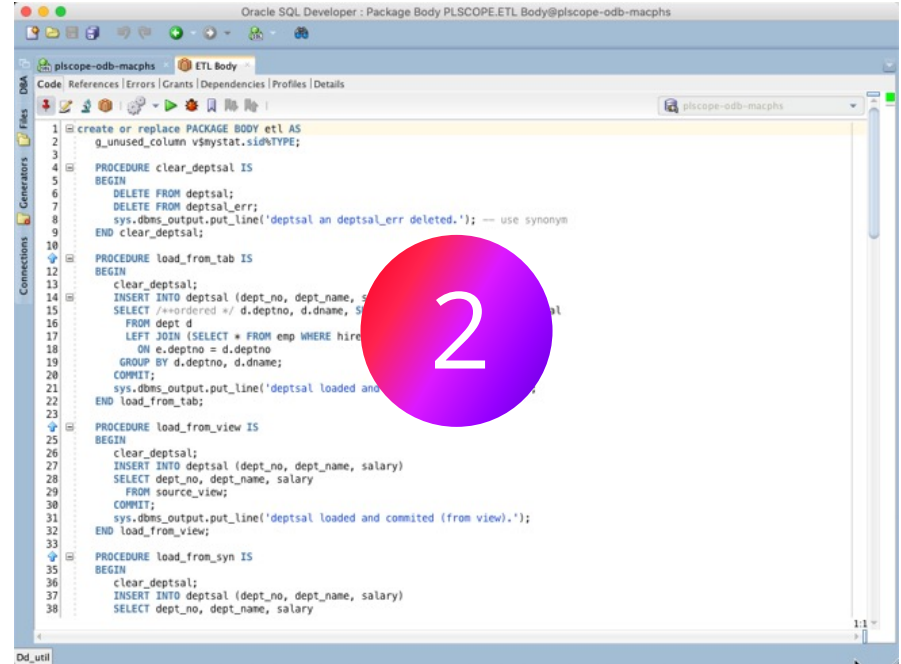
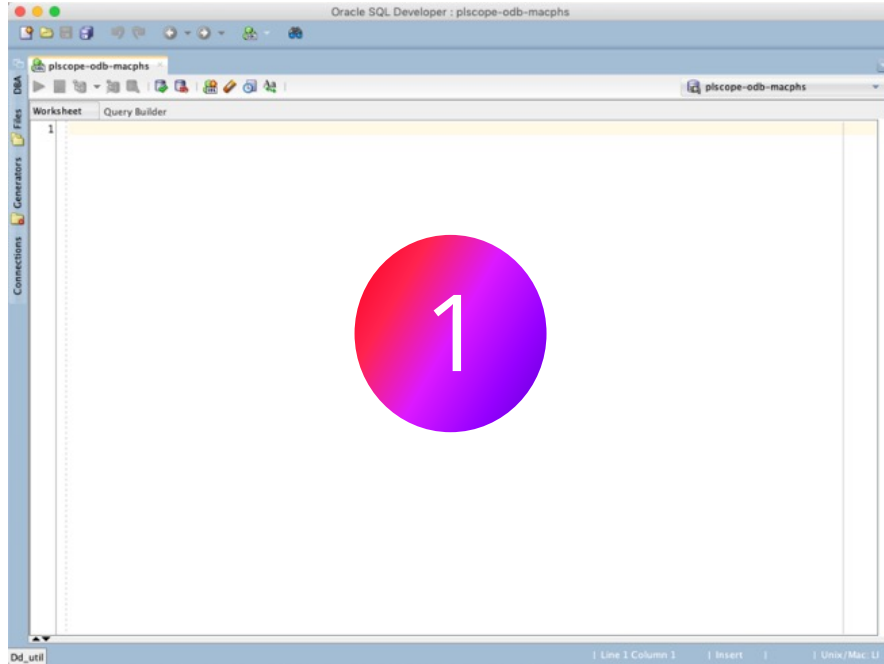
- Download utplsqli_for_SQLDev_*.zip from <https://github.com/utPLSQL/utPLSQL-SQLDeveloper/releases>
- Start SQL Developer
- Select "Check for Updates..." in the help menu *)
- Use the "Install From Local File" option to install the previously downloaded "utplsqli_for_SQLDev_*.zip" file
 - User must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



*) You can also configure an Update Center, see <https://github.com/PhilippSalvisberg/sqldev-update>

BUILD & RUN TESTS IN SQL DEVELOPER

14 STARTING POINT?



RUN CODE COVERAGE REPORTS IN SQL DEVELOPER

16 CODE COVERAGE – DEFINITION

"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

17 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

18 CODE BLOCK COVERAGE (12.2 AND HIGHER)

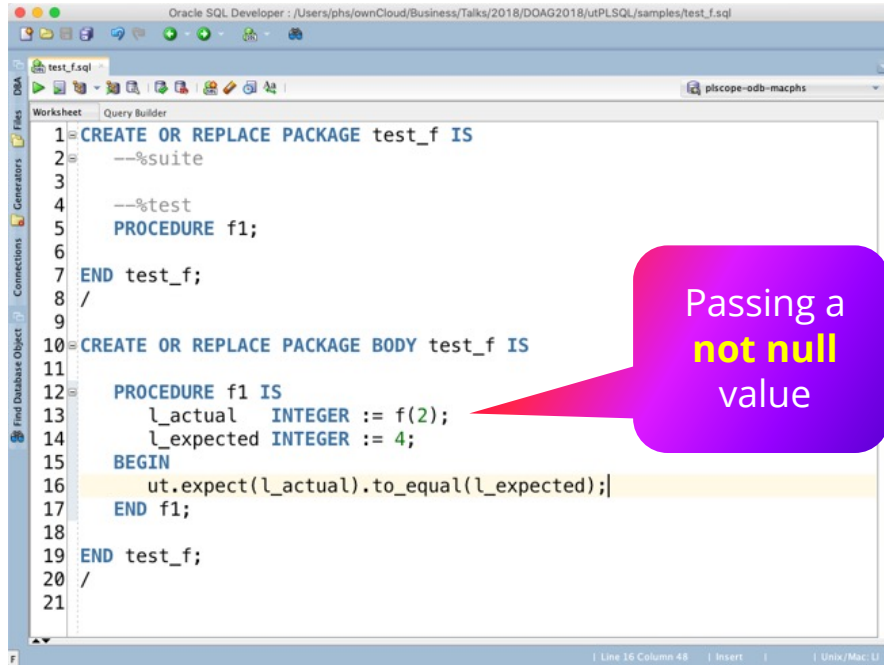
```
CREATE OR REPLACE FUNCTION f(a IN INTEGER) RETURN INTEGER IS  
BEGIN  
    IF a IS NULL THEN RETURN 0; ELSE RETURN a*a; END IF;  
END f;
```

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;
```

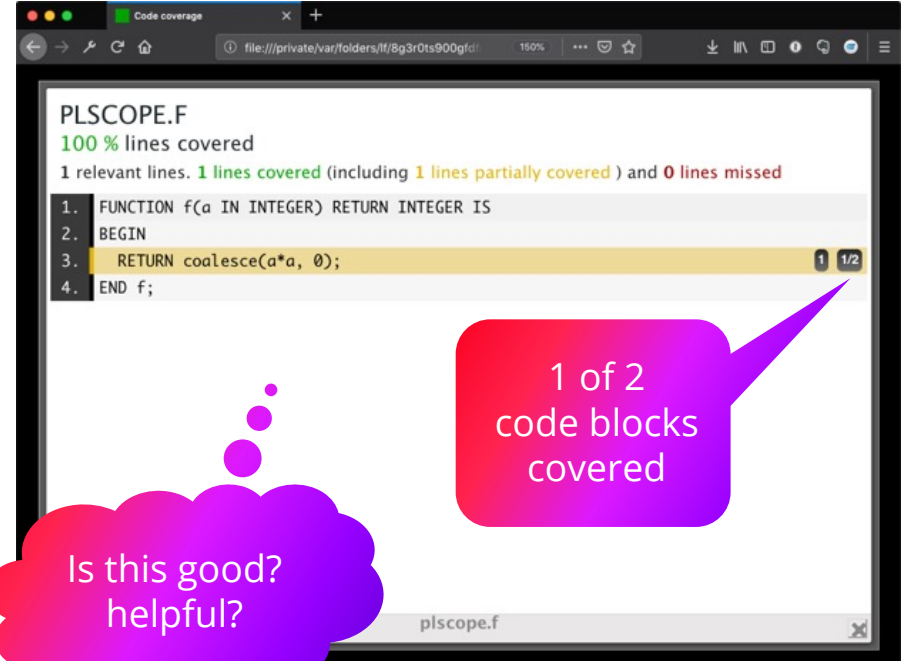
One test case for
100% coverage
when passing NULL

19 utPLSQL – 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?

CORE MESSAGES

21 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

- **utPLSQL will change how you code**

- Write smaller units
- Isolate code that is difficult to test



**TOGETHER WE ARE
#1 PARTNER FOR BUSINESSES TO
HARNESS THE POWER OF DATA
FOR A SMARTER LIFE**