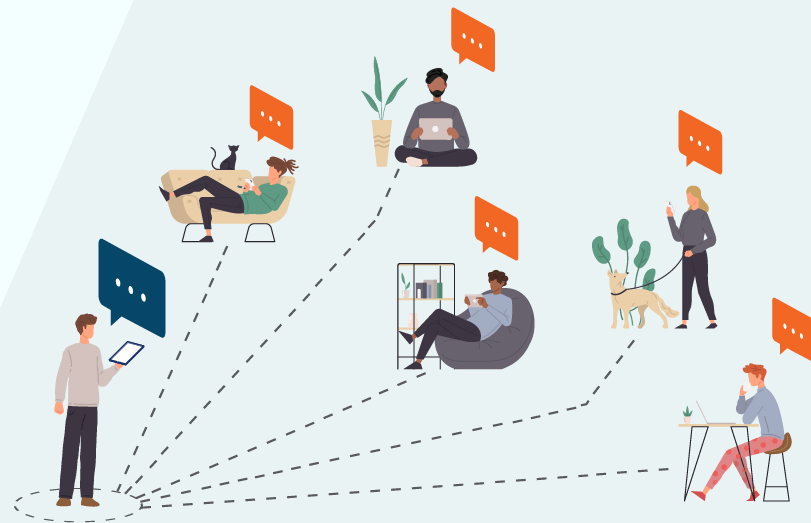




Learn from Home S E R I E S

Tuesday, May 19th - Friday, May 22nd



Using the PL/SQL Debugger

ODTUG – Learn from Home Series

Philipp



@phsalvisberg



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

20.05.2020

Philipp

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



 @phsalvisberg

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

Agenda

trivadis

1. Fundamentals
2. Prerequisites
3. Run & Debug in SQL Developer
4. Run APEX App & Debug in SQL Developer
5. Core Messages

Fundamentals

PL/SQL Debugger in SQL Developer

Oracle SQL Developer : Package Body PLSCOPE.PARSE_UTIL Body@plscope-odb-macphs

Connections | Files | Stack

main (Current, Runnable)

Class Method

- PARSE_UTIL GET_INSERT_TARGETS
- TEST_PARSE_UTIL TEST_GET_INSERT_TARGETS
- ANONYMOUS BLOCK
- DBMS_SQL <procedure\$46>
- UT_EXECUTABLE DO_EXECUTE
- UT_EXECUTABLE DO_EXECUTE
- UT_EXECUTABLE_TEST DO_EXECUTE
- UT_EXECUTABLE_TEST DO_EXECUTE
- UT_TEST DO_EXECUTE
- UT_SUITE_ITEM DO_EXECUTE
- UT_SUITE DO_EXECUTE
- UT_SUITE_ITEM DO_EXECUTE
- UT_LOGICAL_SUITE DO_EXECUTE
- UT_SUITE_ITEM DO_EXECUTE
- UT_LOGICAL_SUITE DO_EXECUTE
- UT_RUN DO_EXECUTE
- UT_SUITE_ITEM DO_EXECUTE
- UT_RUNNER RUN
- UT RUN_AUTONOMOUS
- UT RUN
- UT RUN
- ANONYMOUS BLOCK

Code Details References Dependencies Grants Errors Profiles

plscope-odb-macphs

```
79 FUNCTION get_insert_targets(  
80     in_parse_user IN VARCHAR2,  
81     in_sql        IN CLOB  
82 ) RETURN  
83 IS  
84     t_obj t_obj_type := t_obj_type();  
85     l_xml xmltype;  
86 BEGIN  
87     IF regexp_like(in_sql, '^(\\s*)(INSERT)(.+)$', 'in') THEN  
88         l_xml := parse_query(in_parse_user => in_parse_user, in_query => in_sql);  
89         <<targets>>  
90         FOR r_tar IN (  
91             SELECT schema_name,  
92                    table_name  
93             FROM xmltable(q'{
```

PACKAGE BODY | PROCEDURE | FUNCTION get_insert_targets | BEGIN | IF regexp_like | FOR r_tar | (| 91:1

Debugging: ldeConnections%23plscope-odb-macphs.jpr - Log | Breakpoints | Smart Data | Data | Watches

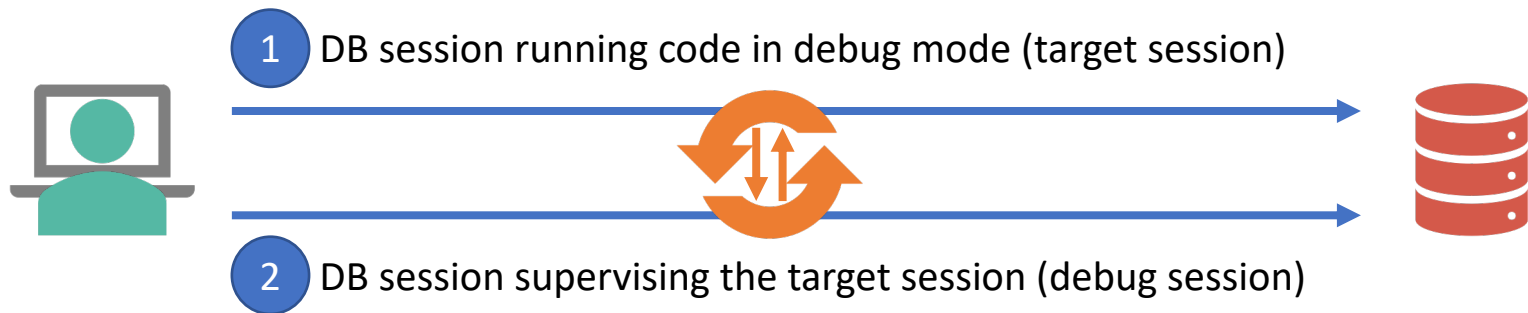
Name Value Type

R_TAR

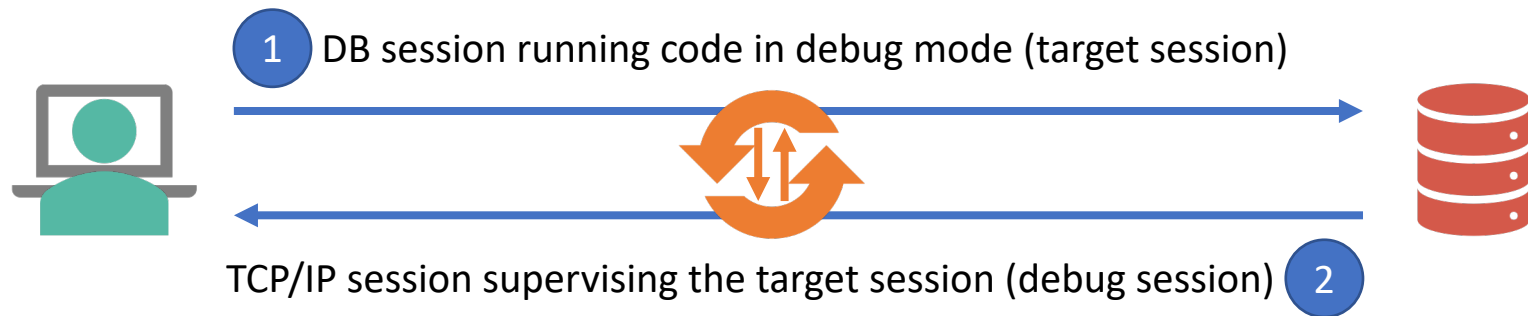
- SCHEMA_NAME NULL VARCHAR2(128)
- TABLE_NAME 'DEPT' VARCHAR2(128)

PL/SQL compiled/saved

DBMS_DEBUG (deprecated since 12.2)



DBMS_DEBUG_JDWP (available since 9.0)



Comparison

Topic	DBMS_DEBUG	DBMS_DEBUG_JDWP
TOAD (Quest)	✓	✗
PL/SQL Developer (Allround Automations)	✓	✗
SQL Developer (Oracle)	✓	✓
Debug PL/SQL	✓	✓
Debug Java Stored Procedures	✗	✓
Debug Java and PL/SQL (same session)	✗	✓
Remote Debugging (e.g. from APEX applications)	✗	✓



New in SQL Developer 20.2: Select Debugging Package in Preferences

Prerequisites

Compile for Debug

- Set environment (PLSQL_DEBUG is deprecated)

```
ALTER SESSION SET plsql_optimize_level = 1;
```

- Compile all PL/SQL code in current schema

```
BEGIN
  FOR r IN ( -- not handled by dbms_utility.compile_schema
    SELECT object_name
      FROM user_objects
     WHERE object_type = 'TYPE BODY')
  LOOP
    EXECUTE IMMEDIATE 'ALTER TYPE ' || r.object_name || ' COMPILE BODY';
  END LOOP;
  dbms_utility.compile_schema(
    schema          => USER,
    compile_all      => true,
    reuse_settings   => false
  );
END;
```

Privileges to Debug Own Sessions

- Allows a user to connect his current session to a debugger

```
GRANT DEBUG CONNECT SESSION TO <user>;
```

Privileges to Debug Other Sessions

- Allows a user to connect a session by any logon user to a debugger

```
GRANT DEBUG CONNECT ANY TO <user>;
```

- Allows a user to connect any of the specified user's logon sessions to a debugger to debug another user's session or his own

```
GRANT DEBUG CONNECT ON USER <logon_user> TO <user>;
```

Visibility of Other's Code

- Access to public and nonpublic variables, methods, and types
- Place a breakpoint in procedure, function, package, type, view (trigger) or table (trigger)

```
GRANT DEBUG ON <object_name> TO <user>;
```

- Access to public and nonpublic variables, methods, and types of any PL/SQL code in the DB
- Place breakpoint in any PL/SQL code in the DB

```
GRANT DEBUG ANY PROCEDURE TO <user>;
```

Remote Debugging

- Allow `DBMS_DEBUG_JDWP` to establish a connection over TCP/IP to the debug host

```
BEGIN
  dbms_network_acl_admin.append_host_ace (
    host => '<host_name>',
    ace  => sys.xs$ace_type(
      privilege_list => sys.xs$name_list('JDWP') ,
      principal_name => '<user>',
      principal_type => sys.xs_acl.ptype_db
    )
  );
END;
```

Run & Debug in SQL Developer

Use Case

```
SQL> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> update dept set loc = 'ZURICH' where deptno = 10;
```

1 row updated.

```
SQL> select * from dept where deptno = 10;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK

What?

Demo 1

Run APEX App & Debug in SQL Developer

Grant Privileges

- Access from APEX_PUBLIC_USER to parsing schema OGDEMO
- Avoid "ORA-01031: insufficient privileges"

```
GRANT DEBUG CONNECT ON USER apex_public_user to ogdemo;
```

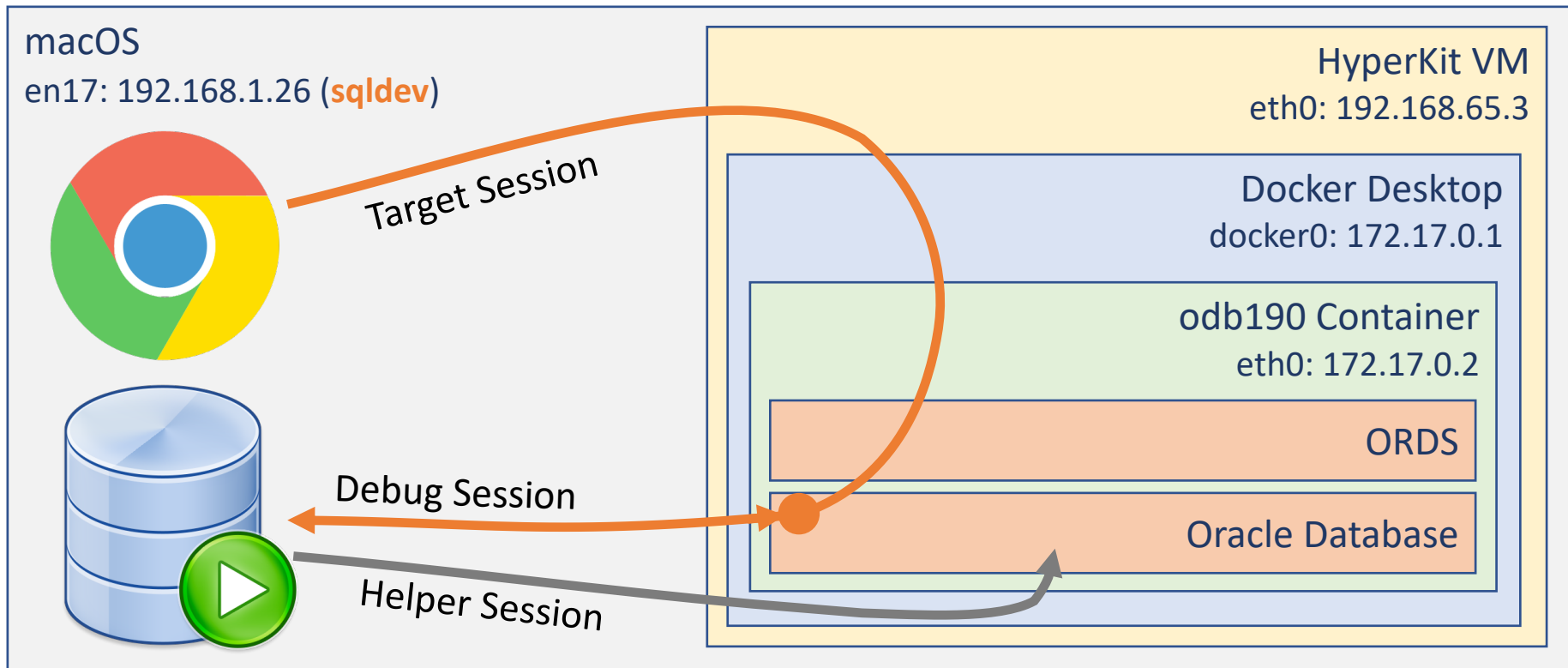
- Access to breakpoints and variables from APEX_PUBLIC_USER
- Necessary to work with the PL/SQL Debugger

```
GRANT DEBUG ANY PROCEDURE to apex_public_user;
```

Does This Work? – No!

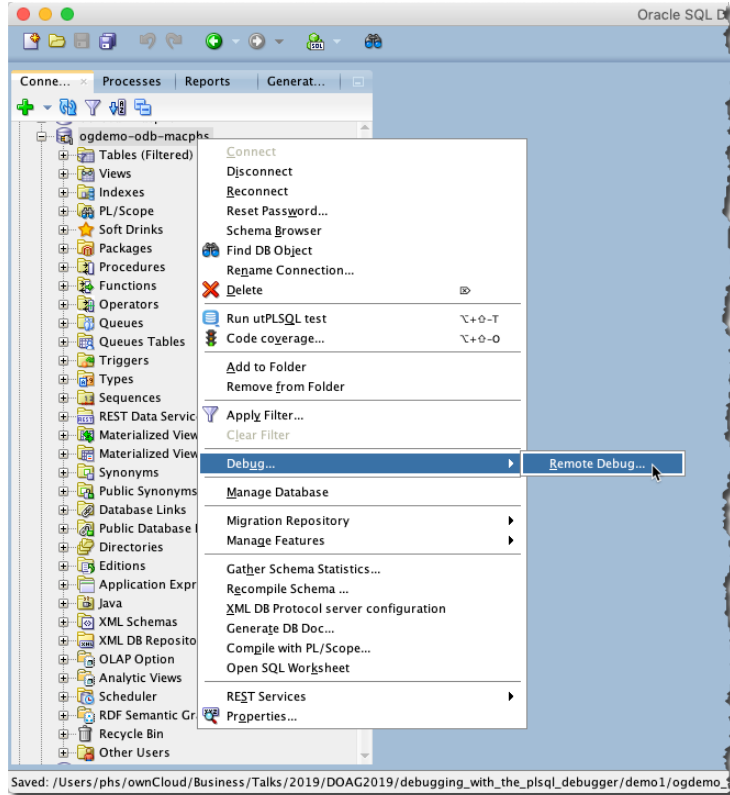
The screenshot shows an Oracle APEX application titled "oddgen Bitemp Remodeler Demo App". The browser address bar displays the URL `localhost:8083/ords/f?p=100:3:345767993:::REMOTE::`, where the `REMOTE::` portion is circled in orange. A thought bubble points to this URL with the text `sys.owa_util.get_cgi_env ('REMOTE_ADDR')`. Another thought bubble points to the application content with the text `parsing schema must be set (www_flow_security_pkg)`. The application interface includes a left-hand navigation menu with items: Home, Non-temporal (expanded), Dept, Emp, Uni-temporal TT, Uni-temporal VT, and Bi-temporal. The main content area, titled "Form on DEPT (2)", contains three input fields: "Deptno" with a red asterisk and the value "10", "Dname" with the value "ACCOUNTING", and "Loc" with the value "ZURICH". At the bottom of the form are "Cancel" and "Apply Changes" buttons.

Connect From Browser to SQL Developer

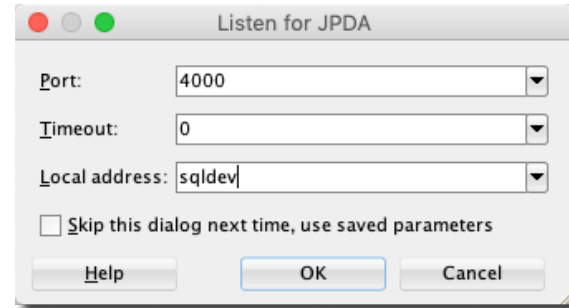


Listen for Incoming Debug Connections

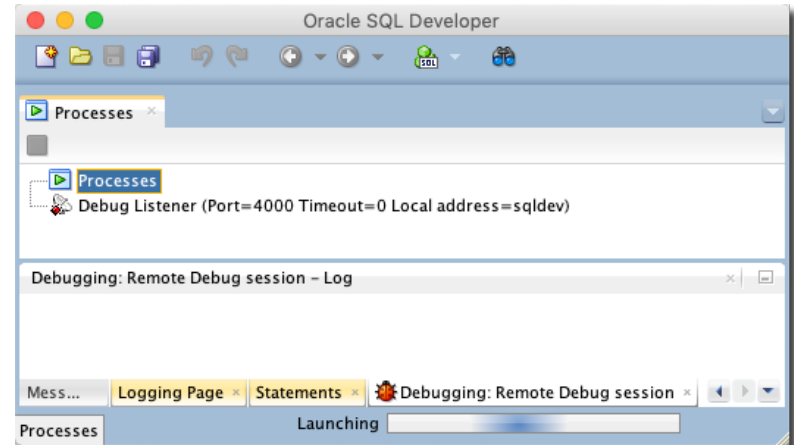
1



2



3



Connect to Debugger from APEX App

The screenshot displays the Oracle APEX App Builder interface. The top navigation bar includes the Oracle logo and tabs for App Builder, SQL Workshop, Team Development, and App Gallery. The main workspace is titled 'Application 100 \ Page Designer'. On the left, a tree view shows the application structure, with 'Processes' expanded and 'start_debug' selected. The central canvas shows a 'Form on DEPT (2)' with three items: P3_DEPTNO, P3_DNAME, and P3_LOC. The right pane is divided into 'Identification' and 'Source' tabs. The 'Source' tab shows the PL/SQL code for the 'start_debug' process, with the line `dbms_debug_jdwp.connect_tcp('sqldev', 4000);` circled in orange.

ORACLE App Builder SQL Workshop Team Development App Gallery

Application 100 \ Page Designer

Computations
Validations
Processes
 Processing
 start_debug
 Process Row of DEPT
 reset page
Branches
 After Processing
 Go To Page 2

Form on DEPT (2)

CONTENT BODY

Form on DEPT (2)

ITEMS

- P3_DEPTNO
- P3_DNAME
- P3_LOC

REGION CONTENT

Process

Filter

Identification

Source

Location Local Database

PL/SQL Code

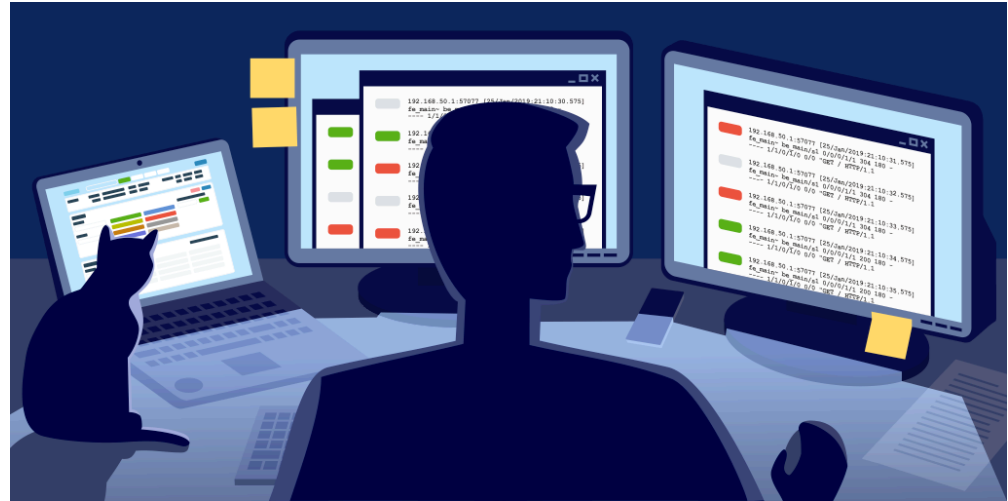
```
declare
    e_not_listening exception;
    pragma exception_init(e_not_listening, -30683);
begin
    dbms_debug_jdwp.connect_tcp('sqldev', 4000);
exception
    when e_not_listening then
        null;
end;
```


Demo 2

Core Messages

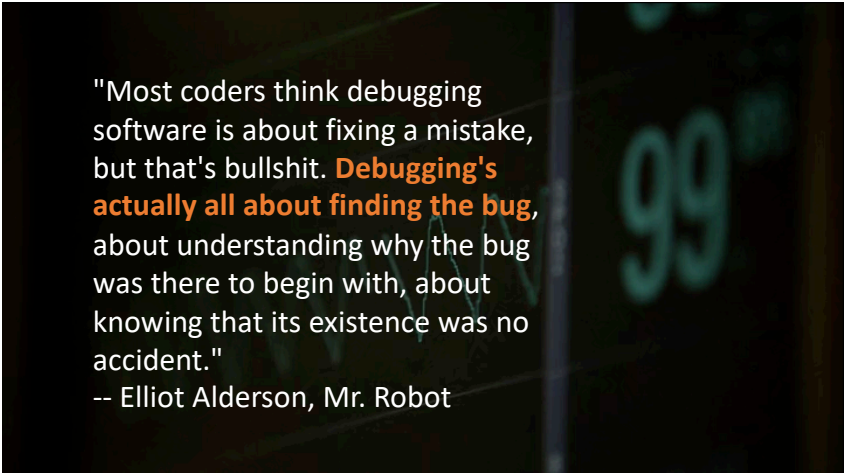
Instrument Your Code

- The PL/SQL debugger is no replacement for log messages in your code
- Look at existing frameworks, such as
 - [APEX DEBUG](#)
 - [Logger](#)
- Use log levels consistently
- Change log level filters at runtime
- Avoid ad-hoc Instrumentation



Use the PL/SQL Debugger

- Set up the PL/SQL Debugger in your development environment
- The PL/SQL Debugger is not perfect
 - You cannot change code while debugging
 - Showing or changing values has limitations
 - Expression support is poor
 - Querying data in the target session is not possible in SQL Developer
- But it helps
 - Finding bugs
 - Understanding the code



"Most coders think debugging software is about fixing a mistake, but that's bullshit. **Debugging's actually all about finding the bug,** about understanding why the bug was there to begin with, about knowing that its existence was no accident."

-- Elliot Alderson, Mr. Robot

ODTUG | Learn from Home S E R I E S

Thank you for attending!
Please fill out your evaluations.

Don't forget to register for

ODTUG
Kscope21
nashville,tn | june 20-24

www.kscope21.odtug.com

