

Debugging with the PL/SQL Debugger

Philipp



@phsalvisberg



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

18.11.2019

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>

Visit Us at Our Booth on Level 3

25
YEARS
OF TRIVADIS

- Trivadis barista (good coffee from morning to night)
- Birthday cake (daily from 14:00)
- Speed-Sessions at the booth with raffle:
 - **Di 14:45:** Martin Berger *"Oracle Cloud Free Tier – einmal kurz ausprobiert..."*
 - **Mi 10:45:** Guido Schmutz *"Warum ich Apache Kafka als IT-Spezialist kennen sollte!"*
 - **Mi 14:45:** Stefan Oehrli *"PDB Sicherheit und Isolation, Herausforderungen von Oracle Container DB's"*
 - **Do 10:45:** Chris Antognini *"Welche Query-Optimizer Funktionalitäten sind nur in Autonomous Database verfügbar? "*
- Participation in our DOAG raffle
- Networking and discussions with our speakers



Agenda

trivadis

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

Motivation

FOEX

APEX Debugging - 101



Peter Raganitsch

CEO FOEX, Oracle ACE Director



Tools

- APEX Debug
- Browser Developer Tools
- SQL Developer Debugging PL/SQL
- Logfiles
- Rubber Duck

FOEX

www.foex.at

See: <https://www.doag.org/de/home/news/aufgezeichnet-peter-raganitsch-apex-debugging-101/detail/>

"Debugging" **What Do You Think of First?**

Explain Code Line by Line?

trivadis



Change Log Level?

Debug Message Data				
localhost:8083/ords/f?p=4000:939:3794274366728::SET_PAGE_ID:NO:939,RIR::				
Elapsed	Execution	Message	Level	Graph
0.57653	0.00095	<pre>characters=409, round bytes=0 select * from(select a.*,row_number() over (order by null) apx\$rownum from(select i.*, count(*) over () as APEXSTOTAL_ROW_COUNT from (select (null) "APEX_ROW_PK","DEPTNO","DNAME","LOC","APEX\$ROWID_ALIAS" "APEX\$ROWID_ALIAS" from ((select /*+ qb_name(apex\$inner) */d."DEPTNO",d."DNAME",d."LOC",d."ROWID" "APEX\$ROWID_ALIAS" "DEPTNO", "DNAME", "LOC" from "OGDEMO"."DEPT" order by deptno) d)) i) i where i=1 and rownum<=1000001)a)where apx\$rownum<=:p5_max_rows</pre>	9	
0.57748	0.00395	parse_as_user p_cursor=>58045293,p_username=>OGDEMO,p_lfflg=>false,p_use_roles=>false	8	
0.58144	0.00077	re_execute_query	8	
0.58220	0.09829	execute	8	
0.68050	0.00103	<pre>SQL_ID fhn7hbussq104, child number 1 select * from(select a.*,row_number() over (order by null) apx\$rownum from(select i.*, count(*) over () as APEXSTOTAL_ROW_COUNT from (select (null) "APEX_ROW_PK","DEPTNO","DNAME","LOC","APEX\$ROWID_ALIAS" "APEX\$ROWID_ALIAS" from ((select /*+ qb_name(apex\$inner) */d."DEPTNO",d."DNAME",d."LOC",d."ROWID" "APEX\$ROWID_ALIAS" from (select "ROWID", "DEPTNO", "DNAME", "LOC" from "OGDEMO"."DEPT" order by deptno) d)) i) i where i=1 and rownum<=1000001)a)where apx\$rownum<=:p5_max_rows Plan hash value: 3540100320 ----- Id Operation Name Rows Bytes Cost (CPU) Time ----- 0 SELECT STATEMENT 2 (100) * 1 VIEW 2 (0) 00:00:01 * 2 WINDOW NOSORT STOPKEY 4 228 2 (0) 00:00:01 3 VIEW 4 228 2 (0) 00:00:01 4 WINDOW BUFFER 4 168 2 (0) 00:00:01 * 5 COUNT STOPKEY 6 VIEW 4 168 2 (0) 00:00:01 7 TABLE ACCESS BY INDEX ROWID DEPT 2 (0) 00:00:01 8 INDEX FULL SCAN PK_DEPT 4 1 (0) 00:00:01 ----- Predicate Information (identified by operation id): ----- </pre>	9	

Add Log Entries?

```

79 FUNCTION get_insert_targets(
80     in_parse_user IN VARCHAR2,
81     in_sql         IN CLOB
82 ) RETURN t_obj_type
83 IS
84     t_obj t_obj_type := t_obj_type();
85     l_xml xmltype;
86 BEGIN
87     dbms_output.put_line('in_sql: ' || i);
88     IF regexp_like(in_sql, '^(\s*)(INSERT)(.+)$', 'in') THEN
89         l_xml := parse_query(in_parse_user => in_parse_user, in_q
90         <<targets>>
91         FOR r_tar IN (
92             SELECT schema_name,
93                    table_name
94             FROM xmltable(q'{
95                                     for $star in /QUERY/FROM/FROM_ITEM
96                                     return
97                                     <target>
98                                     <schemaName>{$star/SCHEMA/text
99                                     <tableName>{$star/TABLE/text(
100                                     </target>
101             }'
102         )

```

Run Tests?

The screenshot displays the Oracle SQL Developer interface. The top window shows the test results for the package body `PLSCOPE.TEST_PARSE_UTIL`. The test `test_get_insert_targets` is highlighted in blue, indicating it passed. The test results table is as follows:

Description (plscope.test.test_parse_util.test_)	Time
parse_query	22 ms
get_insert_targets	91 ms
get_insert_subquery	37 ms
get_dep_cols	13 ms

Below the table, the test details for `test_get_insert_targets` are shown:

Test	Failures	Errors	Warnings	Info
Owner	PLSCOPE			
Package	test_parse_util			
Procedure	test_get_insert_targets			
Description				
Suitepath	plscope.test.test_parse_util.test_get_insert_targets			
Start	2019-10-30 13:41:53.375			

The right window shows the PL/SQL code for the procedure `test_get_insert_targets`. The code is as follows:

```
33 -- test_get_insert_targets
34 --
35 PROCEDURE test_get_insert_targets IS
36   l_actual  t_obj_type;
37   l_expected t_obj_type;
38 BEGIN
39   -- single table insert
40   l_expected := t_obj_type(
41     obj_type(NULL, NULL, 'DEPT')
42   );
43   l_actual := parse_util.get_insert_targets(
44     USER,
45     q'[
46       INSERT INTO dept VALUES (50, 'TRAINING', 'ZURICH')
47     ]'
48   );
49   ut.expect(anydata.convertCollection(l_actual))
50     .to_equal(anydata.convertCollection(l_expected));
51   -- multitable insert
52   l_expected := t_obj_type(
53     obj_type(NULL, NULL, 'EMP'),
54     obj_type(NULL, NULL, 'DEPT')
55   );
56   l_actual := parse_util.get_insert_targets(
```

Set a Breakpoint?

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 IN_SQL  
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'  
94
```

Debugging: IdeConnections%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

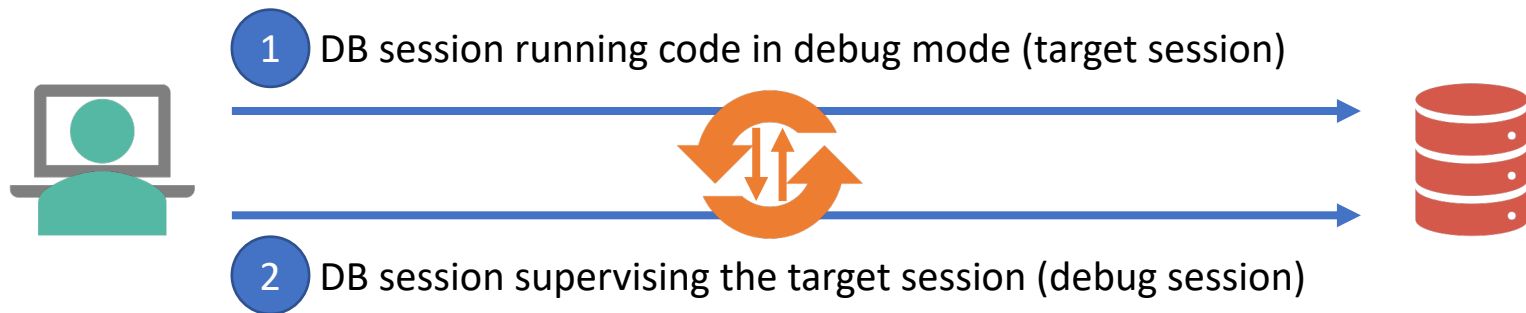
Yes, But...

trivadis

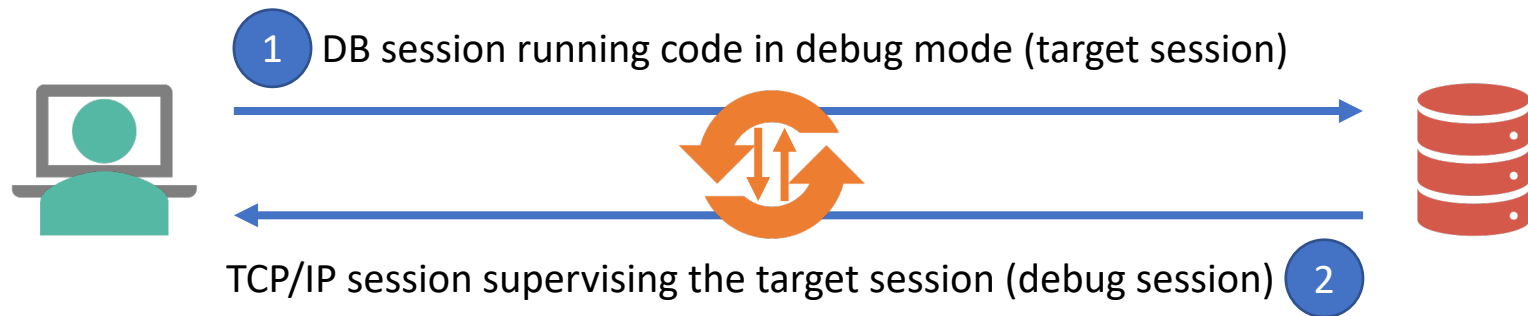


Fundamentals

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)	✗	✓

Prerequisites

Compile for Debug

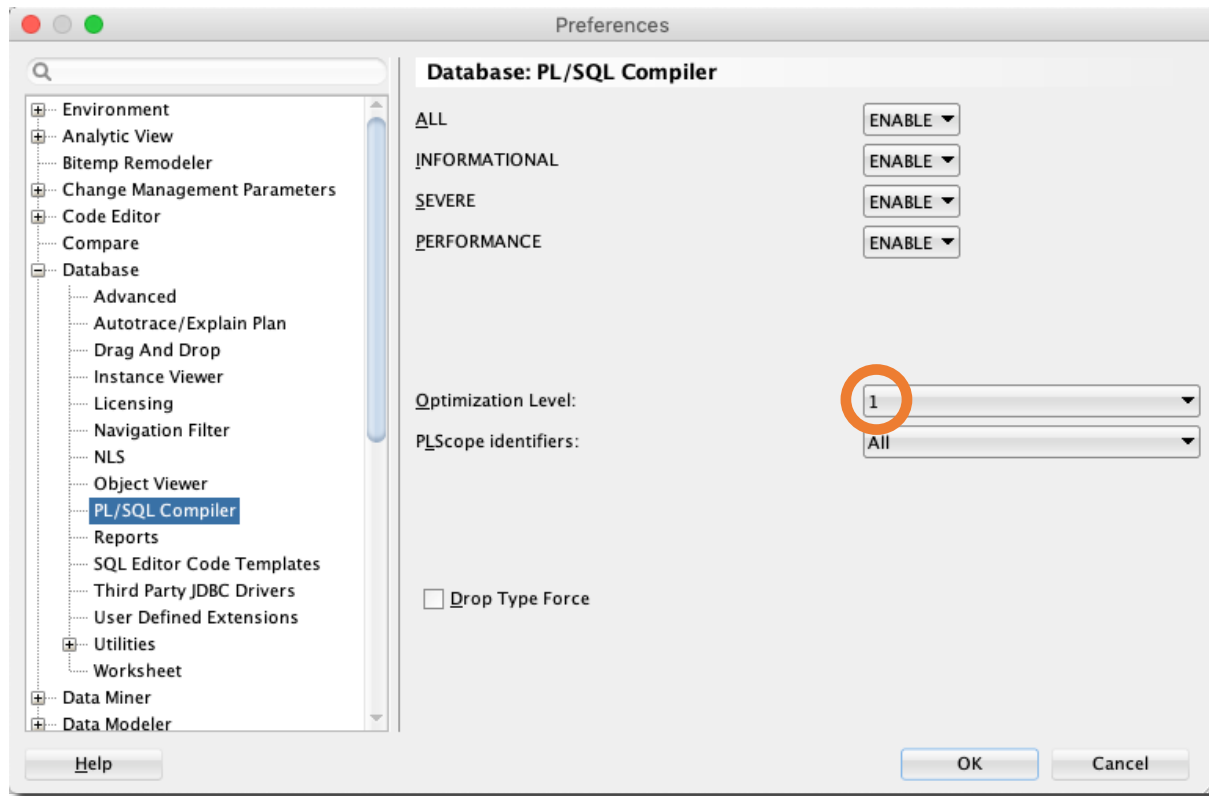
- Set environment (PLSQL_DEBUG is deprecated)

```
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;
```

Compile for Debug in SQL Developer



Privileges to Debug Own Sessions

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

```
GRANT DEBUG CONNECT SESSION TO <debug_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 <debug_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 <target_user> TO <debug_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 <debug_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 <debug_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 => '<target_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.

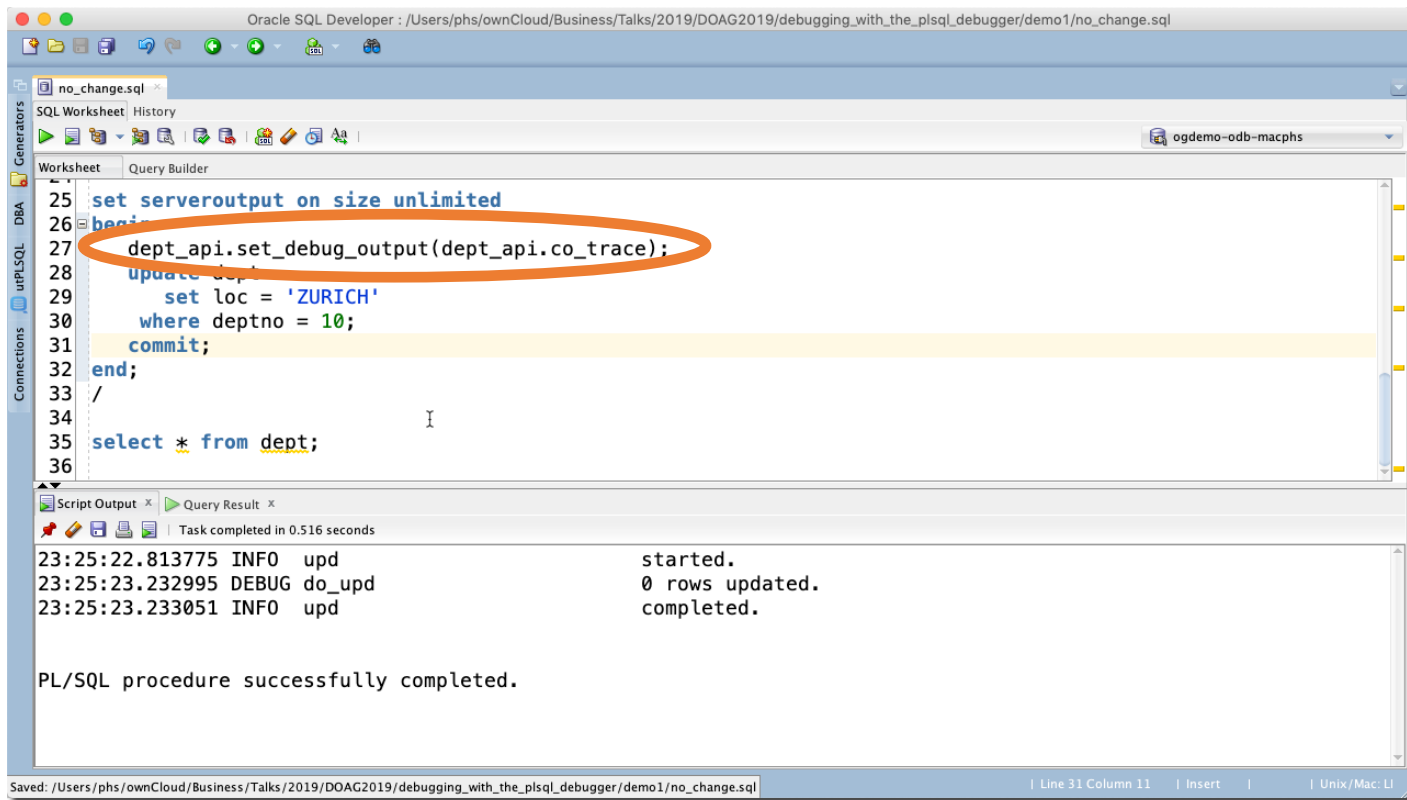
```
SQL> select * from dept where deptno = 10;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK

What?

Demo

Enable Logging



The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in a worksheet. The script is as follows:

```
25 set serveroutput on size unlimited
26 begin
27   dept_api.set_debug_output(dept_api.co_trace);
28   update dept
29     set loc = 'ZURICH'
30     where deptno = 10;
31   commit;
32 end;
33 /
34
35 select * from dept;
36
```

The line `dept_api.set_debug_output(dept_api.co_trace);` is circled in orange. Below the script, the 'Script Output' window shows the execution results:

```
23:25:22.813775 INFO   upd          started.
23:25:23.232995 DEBUG do_upd          0 rows updated.
23:25:23.233051 INFO   upd          completed.

PL/SQL procedure successfully completed.
```

The status bar at the bottom indicates the file is saved at `/Users/phs/ownCloud/Business/Talks/2019/DOAG2019/debugging_with_the_plsql_debugger/demo1/no_change.sql` and shows the cursor is at Line 31, Column 11.

Grant Debug Privileges

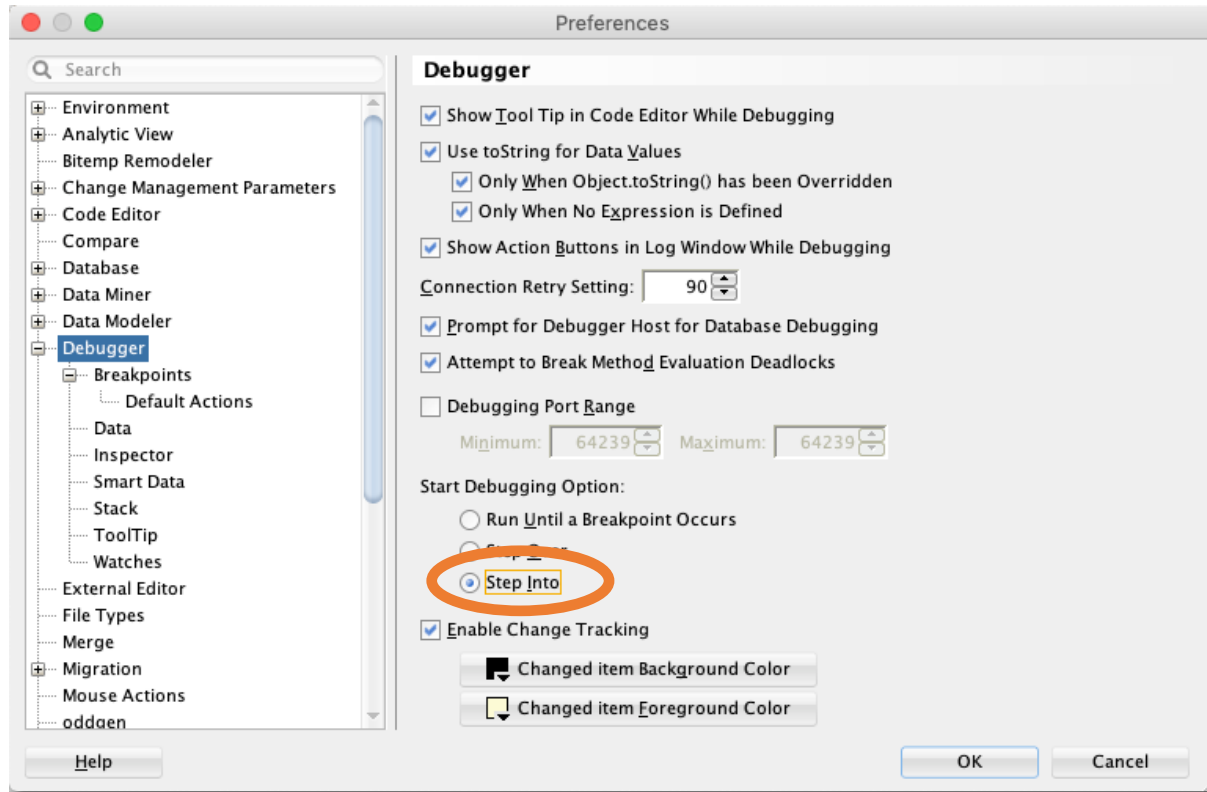
- Debug from ogdemo

```
GRANT DEBUG CONNECT SESSION to ogdemo;
```

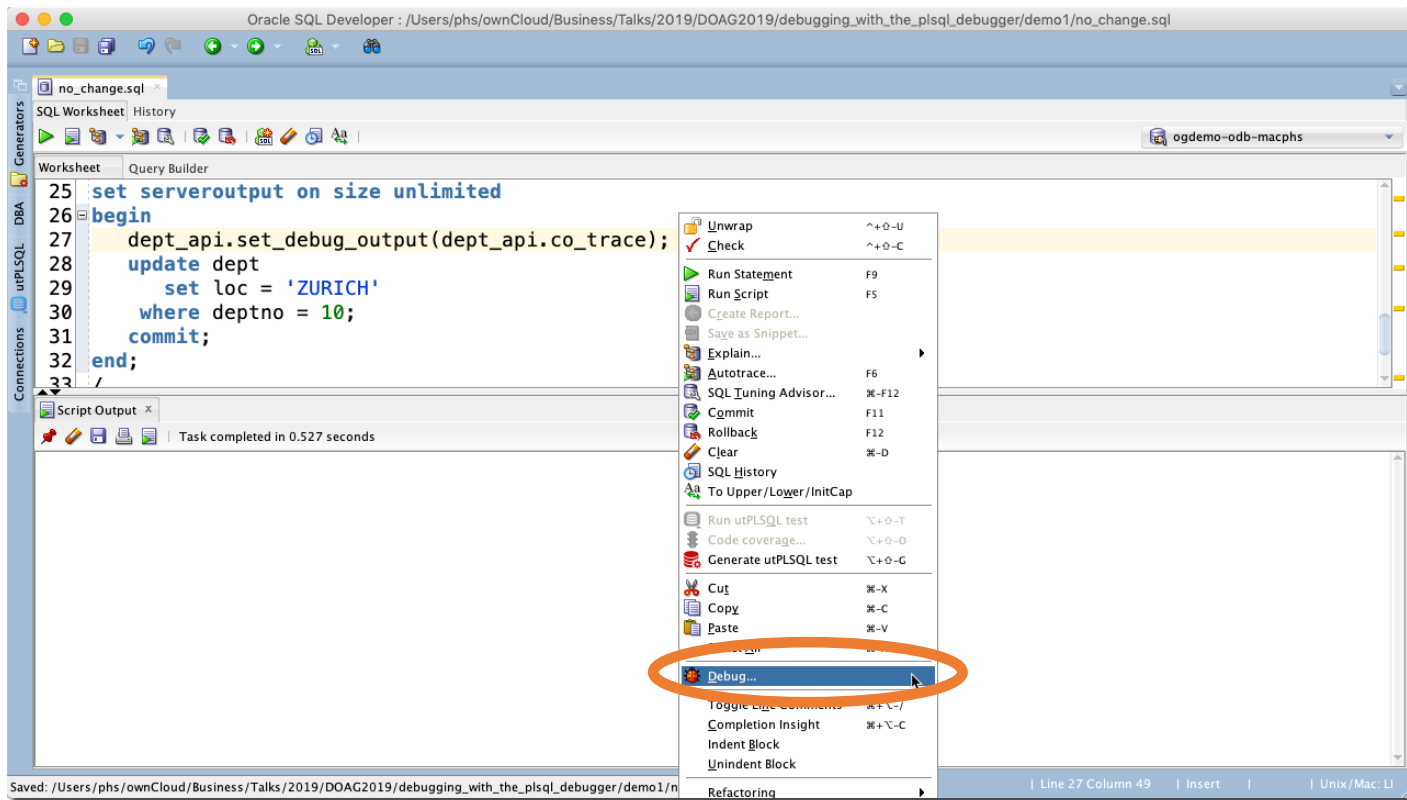
- Access to SQL Developer's Debugger from the Database Server

```
BEGIN
  dbms_network_acl_admin.append_host_ace (
    host => '*',
    ace => sys.xs$ace_type(
      privilege_list => sys.xs$name_list('JDWP') ,
      principal_name => 'ogdemo',
      principal_type => sys.xs_acl.p_type_db
    )
  );
END;
```

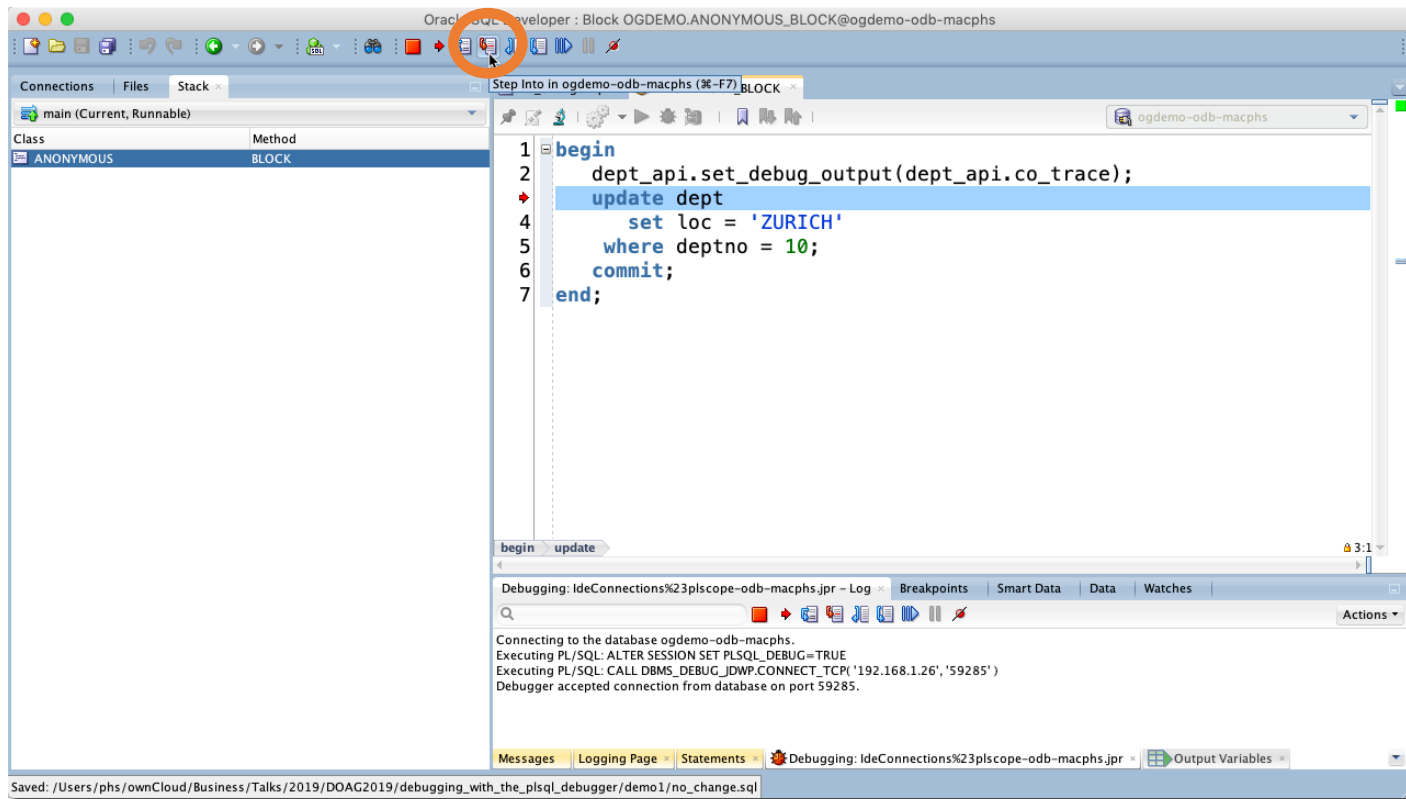
Change Debug Preferences



Debug Anonymous PL/SQL Block



Step Into On Update Statement



Watch Data While Stepping Over/Into

The screenshot shows the Oracle SQL Developer interface with a trigger named `DEPT_TRG` being debugged. The trigger code is as follows:

```
23 l_new_row.deptno := :NEW.deptno;  
24 l_new_row.dname := :NEW.dname;  
25 l_new_row.loc := :NEW.loc;  
26 END IF;  
27  
28 -- Call API  
29  
30 IF INSERTING THEN  
31   dept_api.ins(in_new_row => l_new_row);  
32 ELSIF UPDATING THEN  
33   dept_api.upd(in_new_row => l_new_row, in_old_row => l_old_row);  
34 ELSIF DELETING THEN  
35   dept_api.del(in_old_row => l_old_row);  
36 END IF;  
37 END dept_trg;  
38  
39 TRIGGER dept_trg BEGIN IF INSERTING
```

The watch window at the bottom displays the following data:

Name	Value	Type
L_OLD_ROW		DEPT_OT
L_NEW_ROW		DEPT_OT
DEPTNO	10	NUMBER(2,0)
DNAME	'ACCOUNTING'	VARCHAR2(14)
LOC	'ZURICH'	VARCHAR2(13)
_canBeNull	true	boolean
_isNull	false	boolean
L_VT_START	30.10.19 23:45:54	DATE

The status bar at the bottom indicates the file path: `Saved: /Users/phs/ownCloud/Business/Talks/2019/DOAG2019/debugging_with_the_plsql_debugger/demo1/no_change.sql`

Found

trivadis

The screenshot displays the Oracle SQL Developer interface. The main window shows the code for the `DEPT_HOOK` package body, specifically the `pre_upd` procedure. The code is as follows:

```
18
19
20  PROCEDURE pre_upd (
21      io_new_row IN OUT dept_ot,
22      in_old_row IN dept_ot
23  ) AS
24  BEGIN
25      FOR i in 1..15 LOOP
26          dbms_session.sleep(0.02);
27          IF in_old_row.deptno = i THEN
28              io_new_row.loc := 'NEW YORK';
29          END IF;
30      END LOOP;
31  END pre_upd;
```

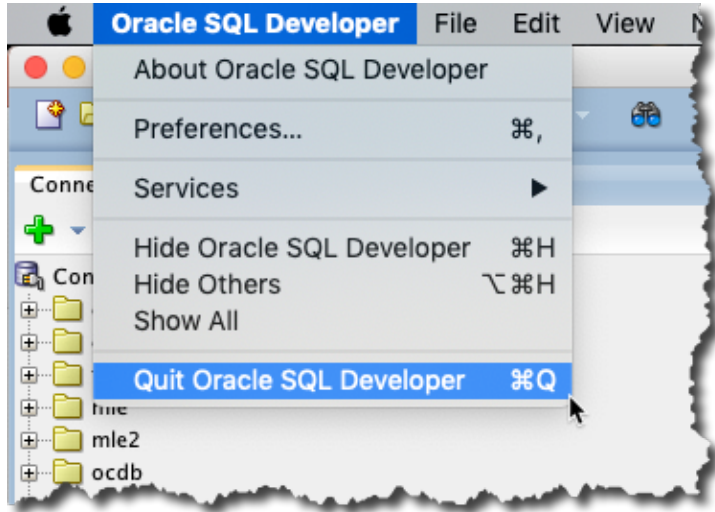
The line `io_new_row.loc := 'NEW YORK';` is highlighted in blue, indicating it is the current statement being debugged. The bottom pane shows the debugging state with the following variables and values:

Name	Value	Type
IO_NEW_ROW		DEPT_OT
IN_OLD_ROW		DEPT_OT
DEPTNO	10	NUMBER(2,0)
DNAME	'ACCOUNTING'	VARCHAR2(14)
LOC	'ZURICH'	VARCHAR2(13)
_canBeNull	true	boolean
_isNotNull	false	boolean
i	10	PLS_INTEGER
Package body variables of DEPT_HOOK		
Package variables of DEPT_HOOK		

The status bar at the bottom indicates the file path: `Saved: /Users/phs/ownCloud/Business/Talks/2019/DOAG2019/debugging_with_the_plsql_debugger/demo1/no_change.sql`.

Using DBMS_DEBUG in SQL Developer

Step 1 – Close SQL Developer

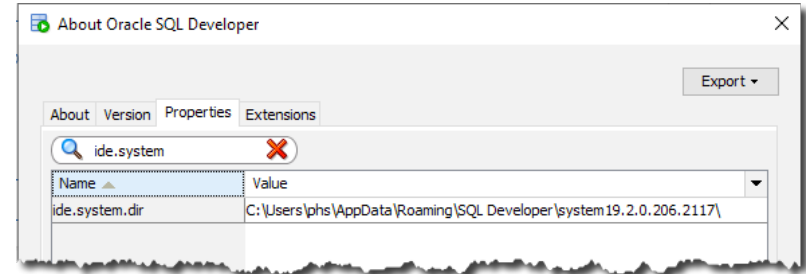
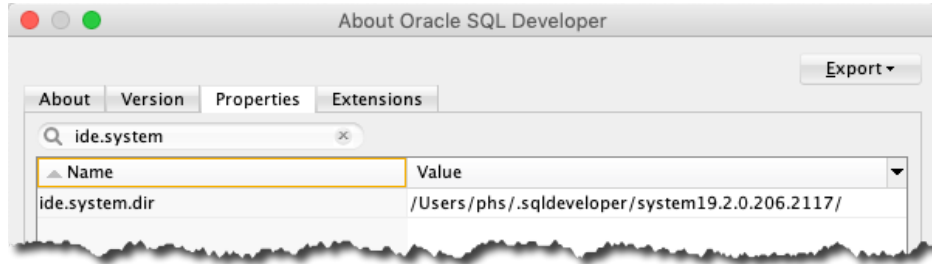


Saves
configuration
files

Ensures, that our
changes on next slide
are not overwritten

Step 2 – Change ide.properties

- Open \${ide.system.dir}/o.sqldeveloper/ide.properties in an editor



- Add the following line

```
DatabaseDebuggerDisableJDWP=true
```

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::`, with the `REMOTE::` portion circled in orange. A thought bubble points to this URL, containing the text `sys.owa_util.get_cgi_env ('REMOTE_ADDR')`. Another thought bubble points to the application content, stating "parsing schema must be set (www_flow_security_pkg)". The application interface includes a left-hand navigation menu with options: 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 a form with the following fields: "Deptno" (marked with a red asterisk, value 10), "Dname" (value ACCOUNTING), and "Loc" (value ZURICH). At the bottom of the form are "Cancel" and "Apply Changes" buttons.

localhost:8083/ords/f?p=100:3:345767993:::REMOTE::

oddgen Bitemp Remodeler Demo App

Home

Non-temporal

Dept

Emp

Uni-temporal TT

Uni-temporal VT

Bi-temporal

Form on DEPT (2)

Deptno * 10

Dname ACCOUNTING

Loc ZURICH

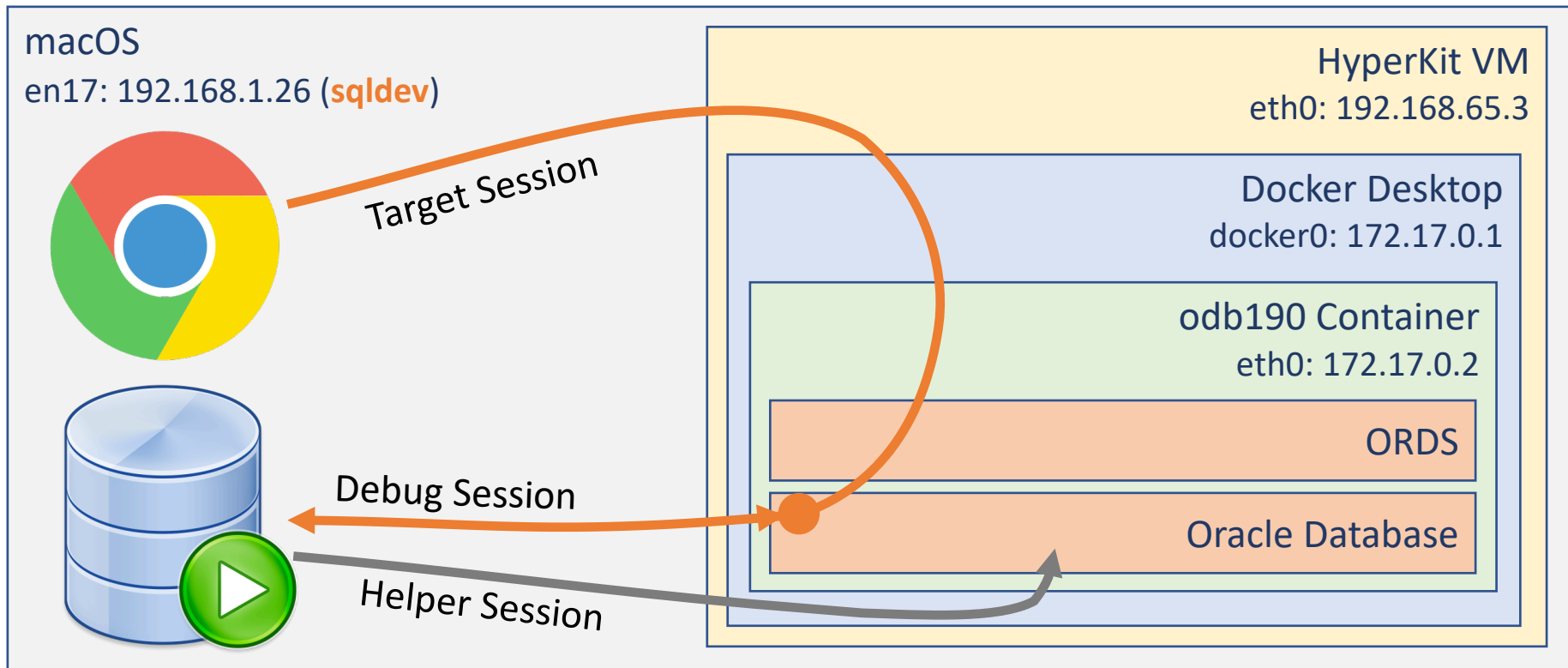
Cancel

Apply Changes

sys.owa_util.get_cgi_env ('REMOTE_ADDR')

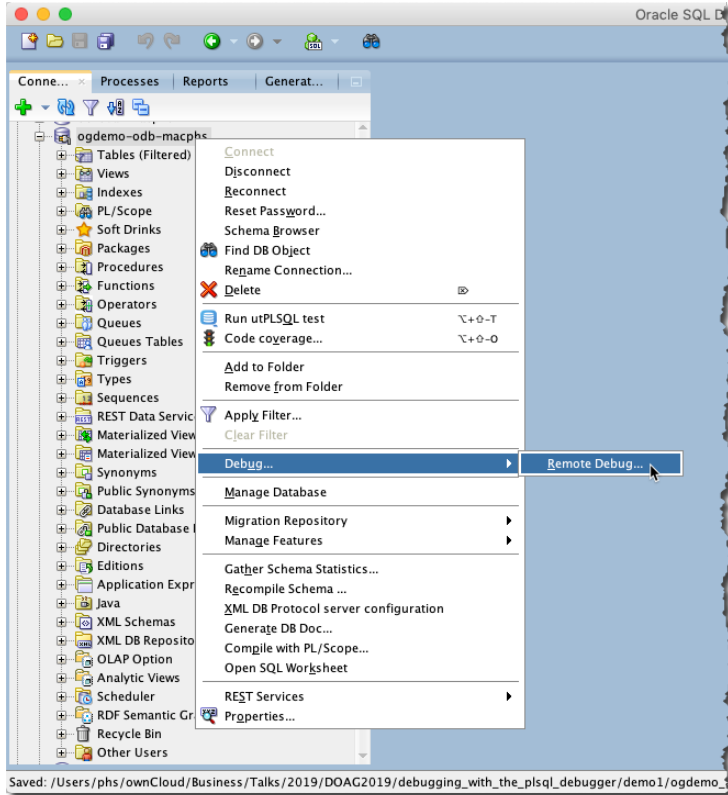
parsing schema must be set (www_flow_security_pkg)

Connect From Browser to SQL Developer

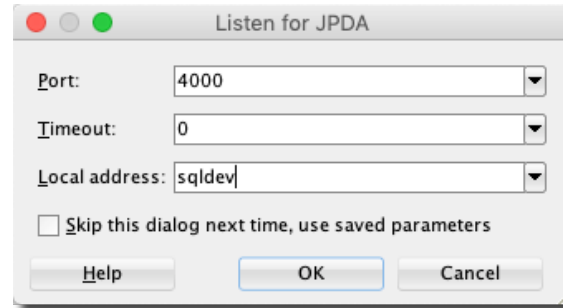


Listen for Incoming Debug Connections

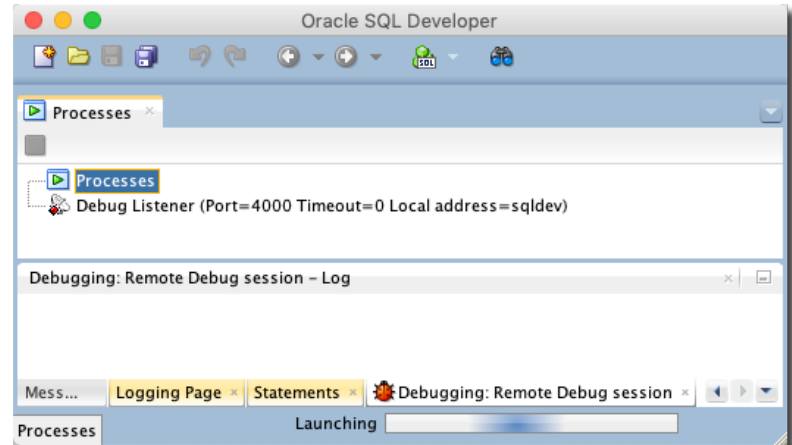
1



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 pane shows the 'Form on DEPT (2)' page, with the 'ITEMS' section containing three items: 'P3_DEPTNO', 'P3_DNAME', and 'P3_LOC'. The right pane shows the 'Source' tab for the 'start_debug' process, containing PL/SQL code. The line `dbms_debug_jdwp.connect_tcp('sqldev', 4000);` is 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

Apply Changes

The screenshot shows a web application interface for 'oddgen Bitemp Remodeler Demo App'. The interface includes a sidebar with navigation links: Home, Non-temporal, Dept, Emp, Uni-temporal TT, Uni-temporal VT, and Bi-temporal. The main content area displays a form titled 'Form on DEPT (2)' with the following fields: Deptno (10), Dname (ACCOUNTING), and Loc (ZURICH). The 'Loc' field is circled in orange. At the bottom right of the form are 'Cancel', 'Delete', and 'Apply Changes' buttons. A mouse cursor is hovering over the 'Apply Changes' button. At the bottom left of the application window, a status bar displays the text 'Waiting for macphs...', which is also circled in orange. The footer of the application window includes the text 'release 1.0 Set Screen Reader Mode On'.

oddgen Bitemp Remodeler Demo App Log Out

Home Non-temporal Dept Emp Uni-temporal TT Uni-temporal VT Bi-temporal

Form on DEPT (2)

Deptno * 10

Dname ACCOUNTING

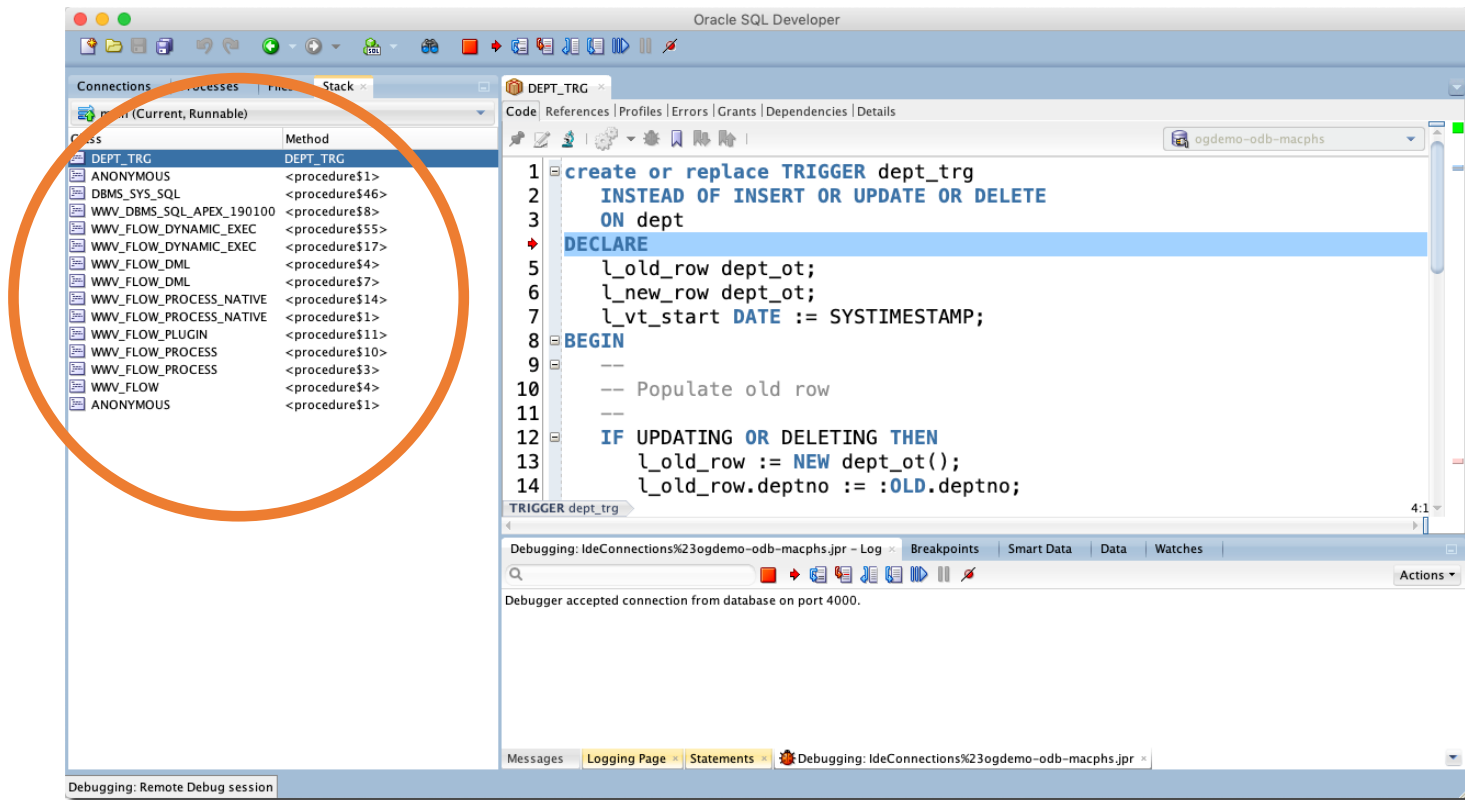
Loc ZURICH

Cancel Delete Apply Changes

release 1.0 [Set Screen Reader Mode On](#)

Waiting for macphs...

Debug in SQL Developer



Debug Session Is Automatically Closed

oddgen Bitemp Remodeler Demo App

Log Out

Home

Non-temporal

Dept

Emp

Uni-temporal TT

Uni-temporal VT

Bi-temporal

Q

Go

Actions

✓

Action Processed.

×

Create

	Deptno	Dname	Loc
	10	ACCOUNTING	NEW YORK
	20	RESEARCH	DALLAS
	30	SALES	CHICAGO
	40	OPERATIONS	BOSTON

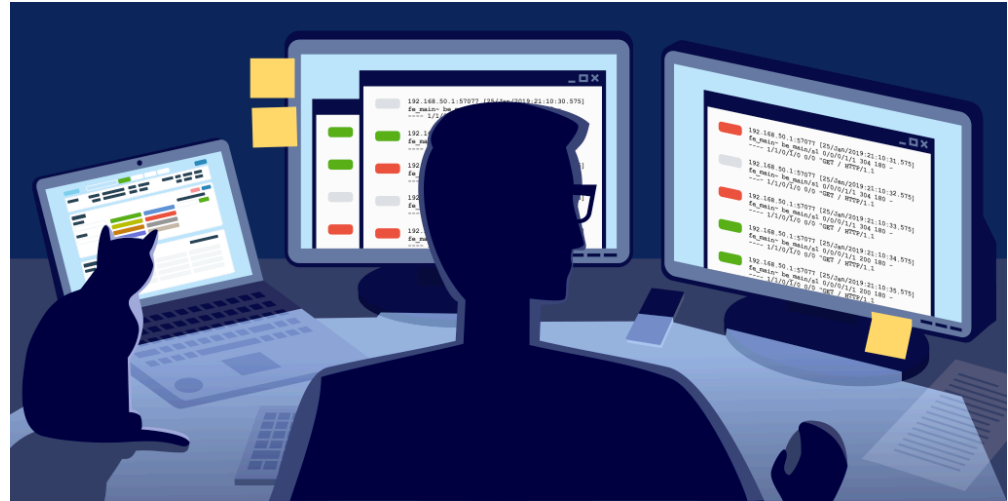
1 - 4

release 1.0 [Set Screen Reader Mode On](#)

Core Messages

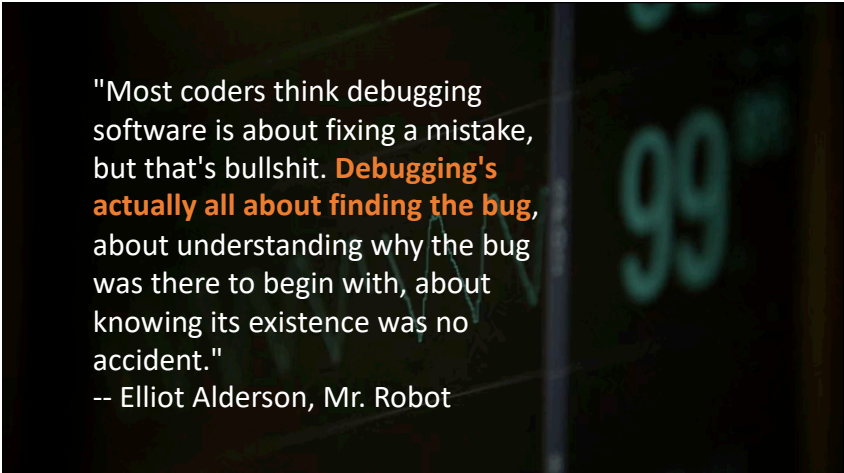
Instrument Your Code

- 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 its existence was no accident."

-- Elliot Alderson, Mr. Robot



Making a **WORLD** possible
in which **intelligent IT**
facilitates **LIFE and WORK** as a
matter of course.