

Debugging with the PL/SQL Debugger

APEX@Home

Philipp

 @phsalvisberg

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

16.04.2020

BASEL | BERN | BRUGG | BUKAREST | DÜSSELDORF | FRANKFURT A.M. | FREIBURG I.BR. | GENF
HAMBURG | KOPENHAGEN | LAUSANNE | MANNHEIM | MÜNCHEN | STUTTGART | WIEN | ZÜRICH



Philipp

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



 @phsalvisberg

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

Agenda

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?



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>select * from(select a.* ,row_number() over (order by null) apx\$rownum from(select i.* , count(*) over () as APEX\$TOTAL_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 1=1 and rownum<=1000001)a)where apx\$rownum<=:p\$_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	

SQL_ID fnh7hbussq104, child number 1

```
select * from(select a.* ,row_number() over (order by null) apx$rownum
from(select i.* , count(*) over () as APEX$TOTAL_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 1=1 and rownum<=1000001 )a)where
apx$rownum<=:p$_max_rows
```

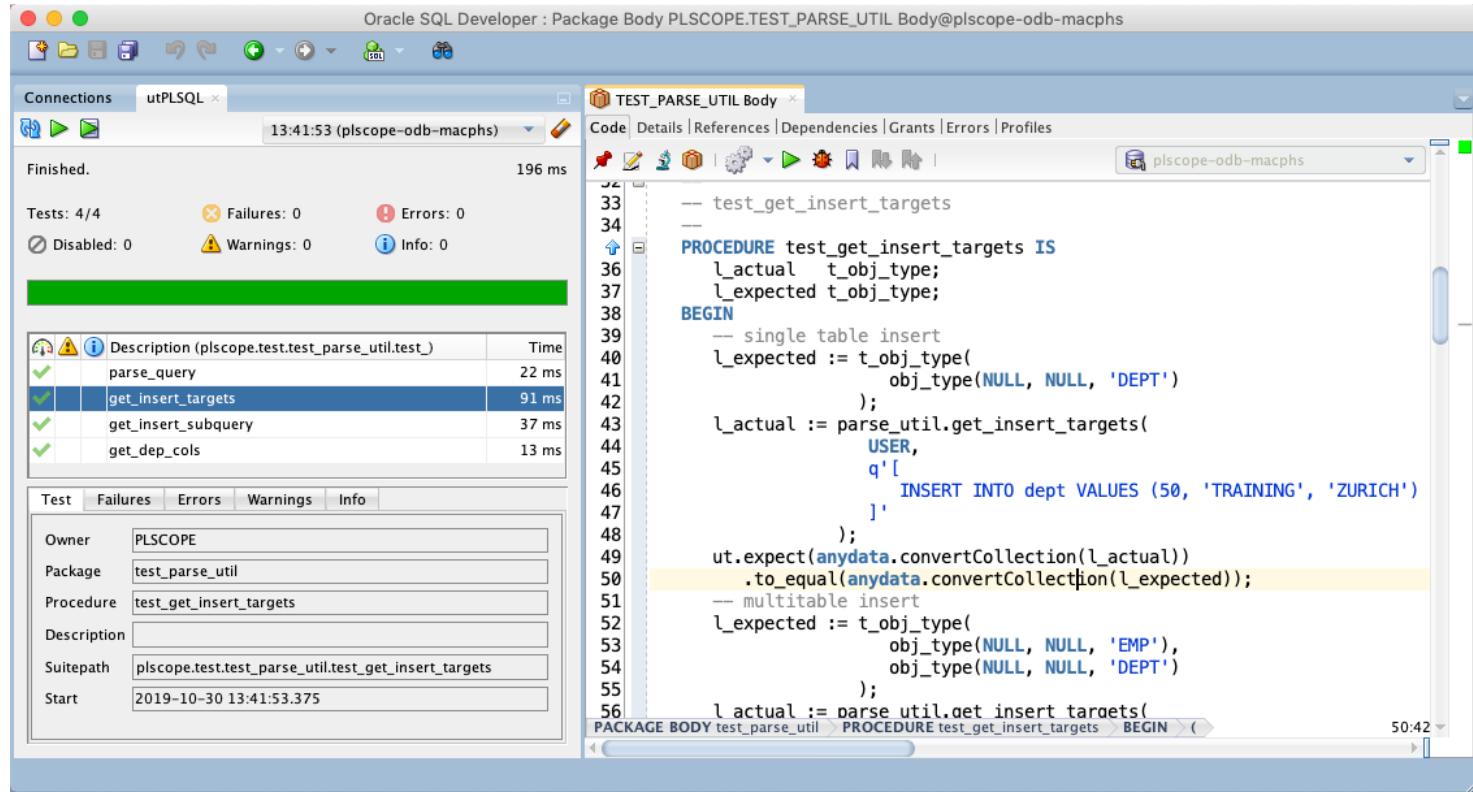
Plan hash value: 3540100320

Id	Operation	Name	Rows	Bytes	Cost (CPU)	Time
0	SELECT STATEMENT				2 (100)	
* 1	VIEW		4	280	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		1	1	1	
6	VIEW		4	168	2 (0)	00:00:01
7	TABLE ACCESS BY INDEX ROWID DEPT	PK_DEPT	4	80	2 (0)	00:00:01
8	INDEX FULL SCAN	PK_DEPT	4	1	(0)	00:00:01

Predicate Information (identified by operation id):

Add Log Entries?

Run Tests?



Set a Breakpoint?

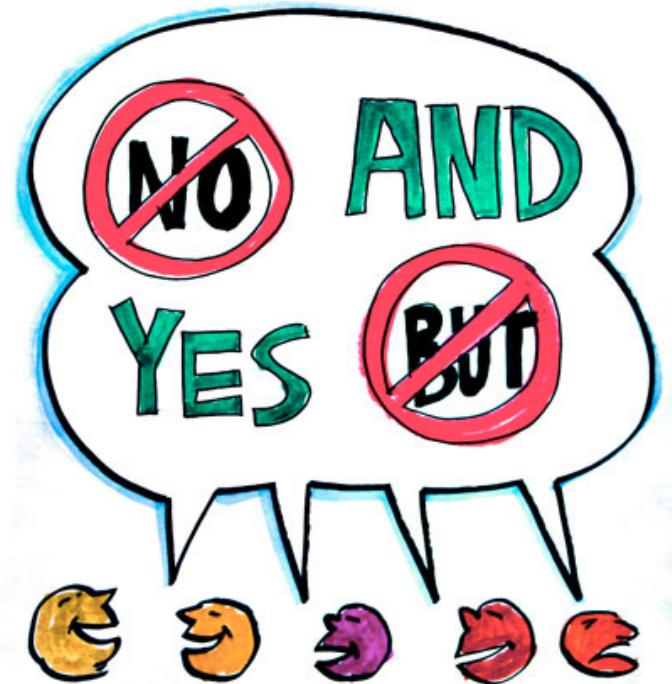
The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : Package Body PLSCOPEPARSE_UTIL Body@plscope-odb-macphs
- Toolbar:** Standard Mac OS X-style toolbar with icons for file operations.
- Connections Tab:** Shows a single connection named "main (Current, Runnable)".
- Code Editor:** Displays the PL/SQL code for the "PARSE_UTIL" package body. The cursor is at line 93, which contains a FOR loop. A red highlight covers the IF...THEN block starting at line 87.
- Watches Tab:** Shows the current values of variables:

Name	Type
R_TAR	VARCHAR2(128)
SCHEMA_NAME	NULL
TABLE_NAME	'DEPT'
- Status Bar:** PL/SQL compiled/saved

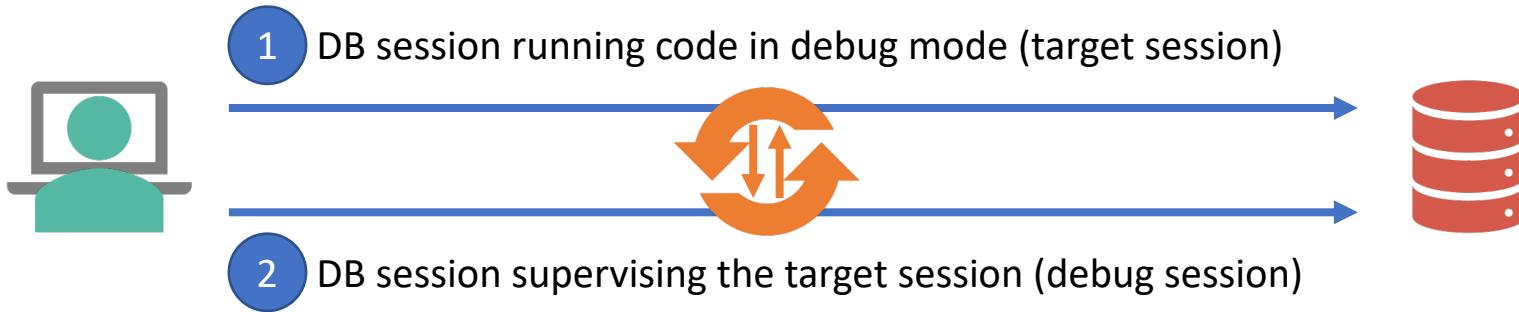
Yes, Setting a Breakpoint, of Course!

trivadis

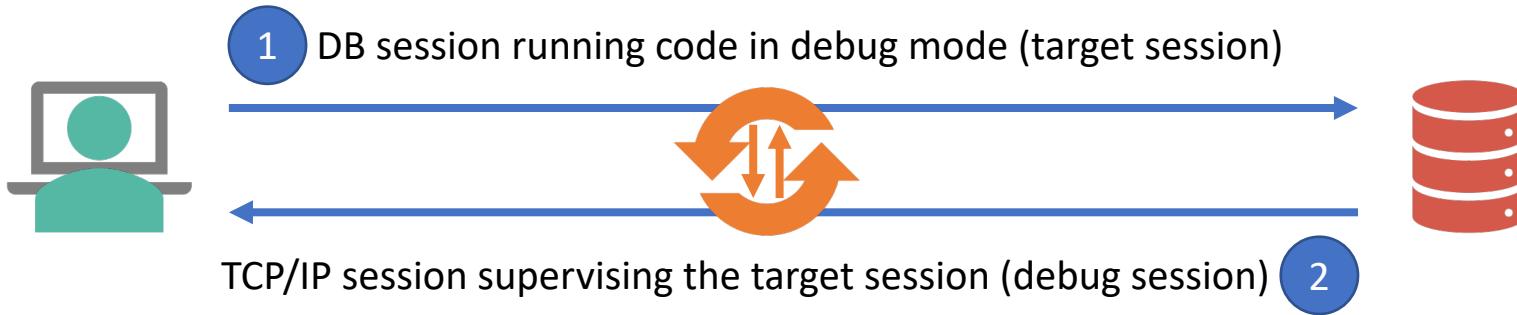


Fundamentals

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)

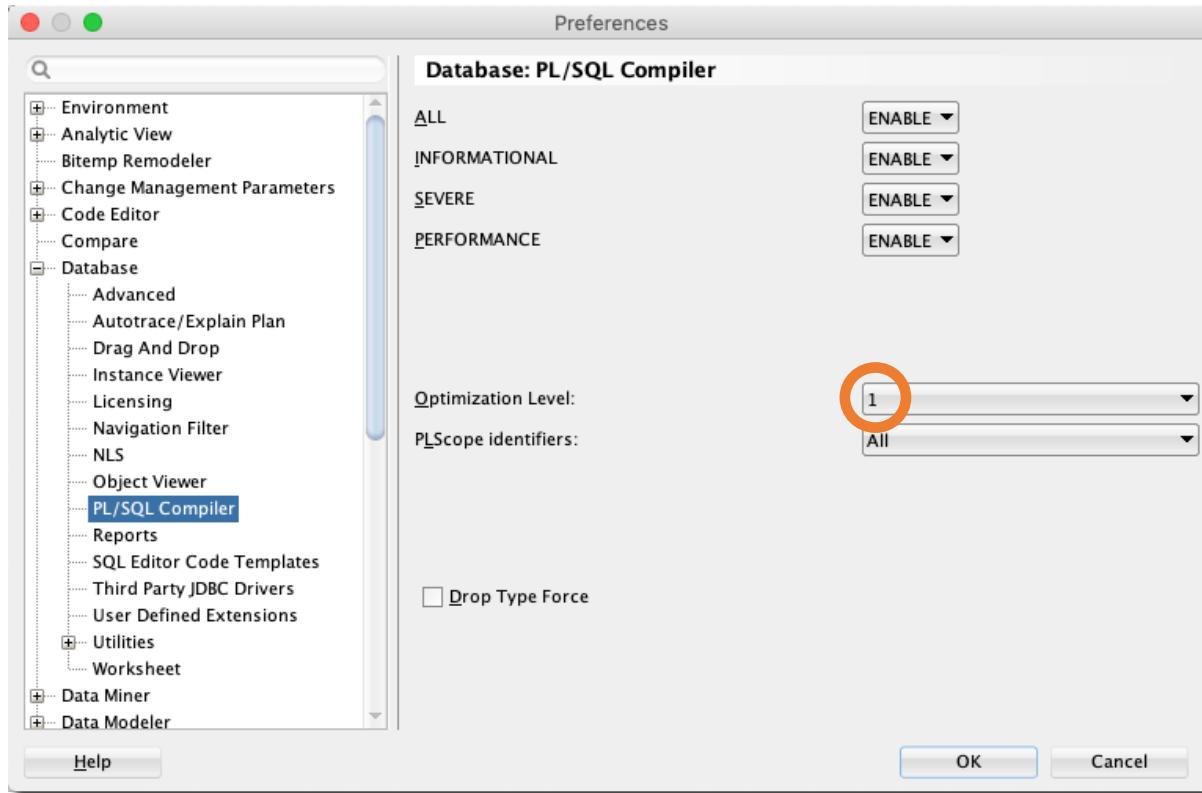
```
ALTER SESSION SET plsql_optimize_level = 1;
```

- 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

trivadis



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;
```

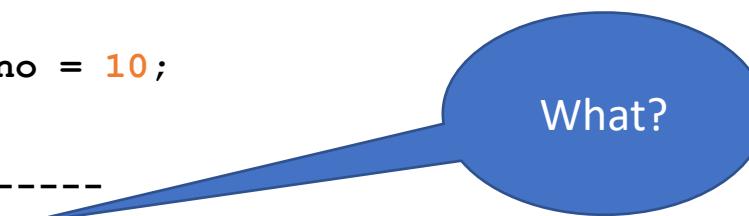
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
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 1

Using DBMS_DEBUG in SQL Developer

Step 1 – Close SQL Developer

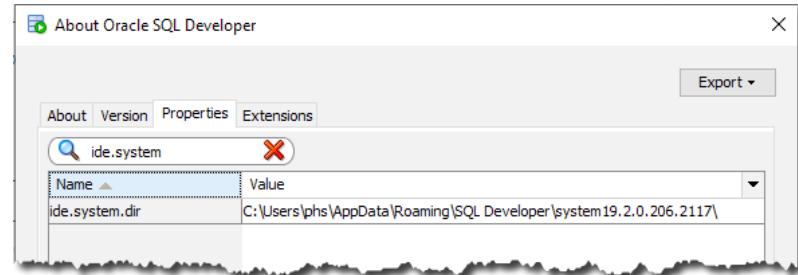
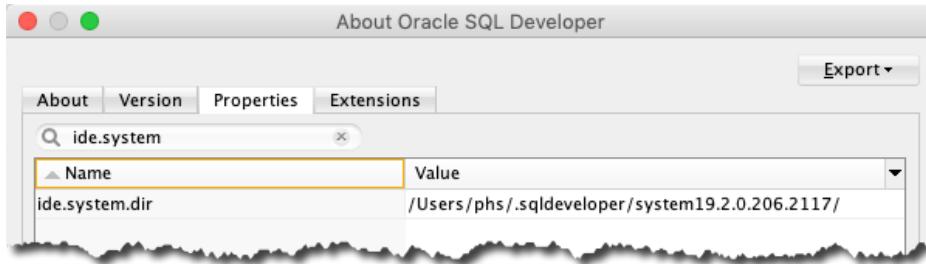


Saves configuration files

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!

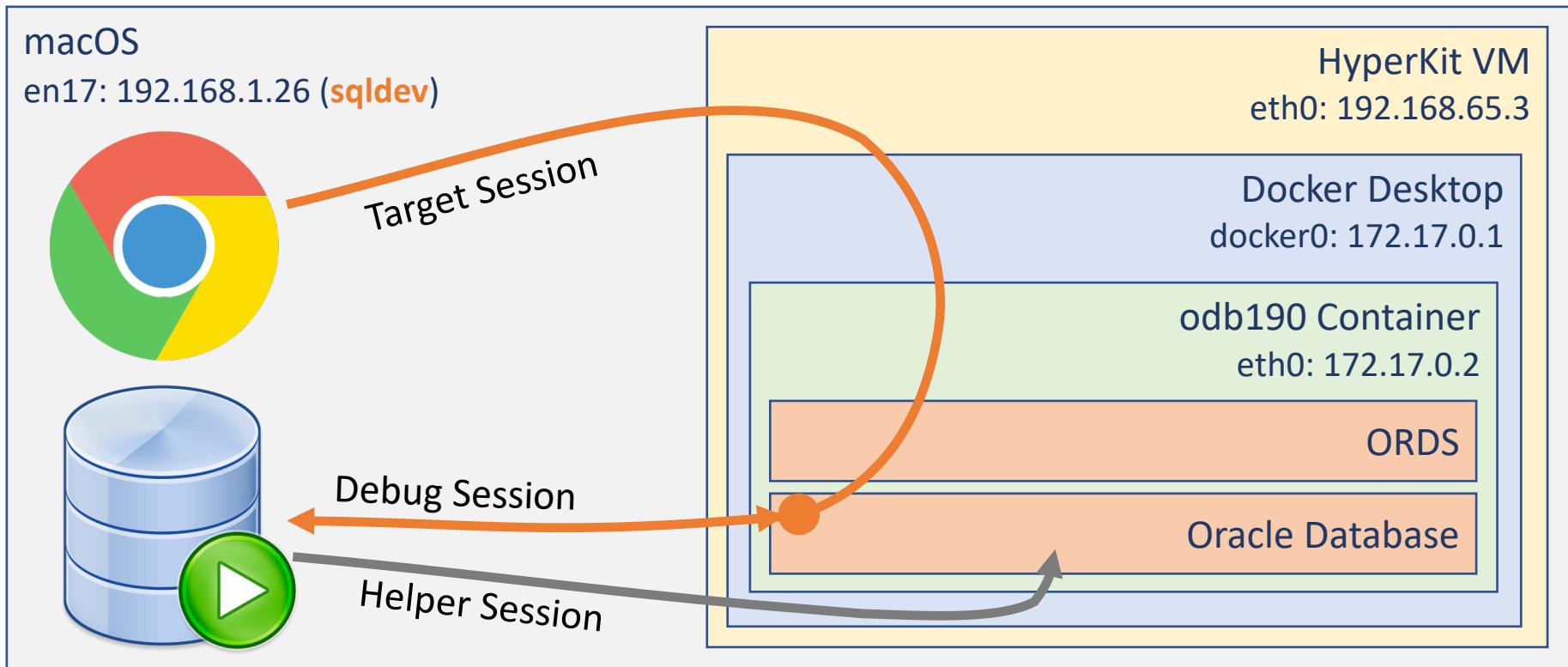
The screenshot shows a web browser displaying an Oracle ADF application titled "oddgen Bitemp Remodeler Demo App". The URL in the address bar is `localhost:8083/ords/f?p=100:3:345767993:257::REMOTE:::`, with the "REMOTE" part highlighted by an orange oval.

The application interface includes a sidebar with navigation items: Home, Non-temporal (Dept, Emp), Uni-temporal TT, Uni-temporal VT, and Bi-temporal. The main content area is titled "Form on DEPT (2)" and contains three input fields: Deptno (10), Dname (ACCOUNTING), and Loc (ZURICH). Below the form are "Cancel" and "Apply Changes" buttons.

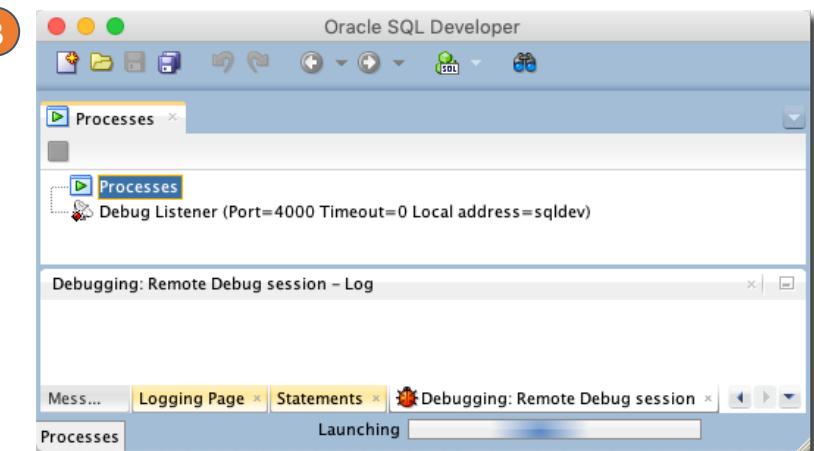
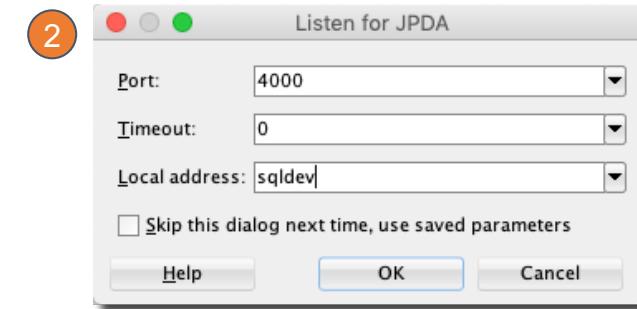
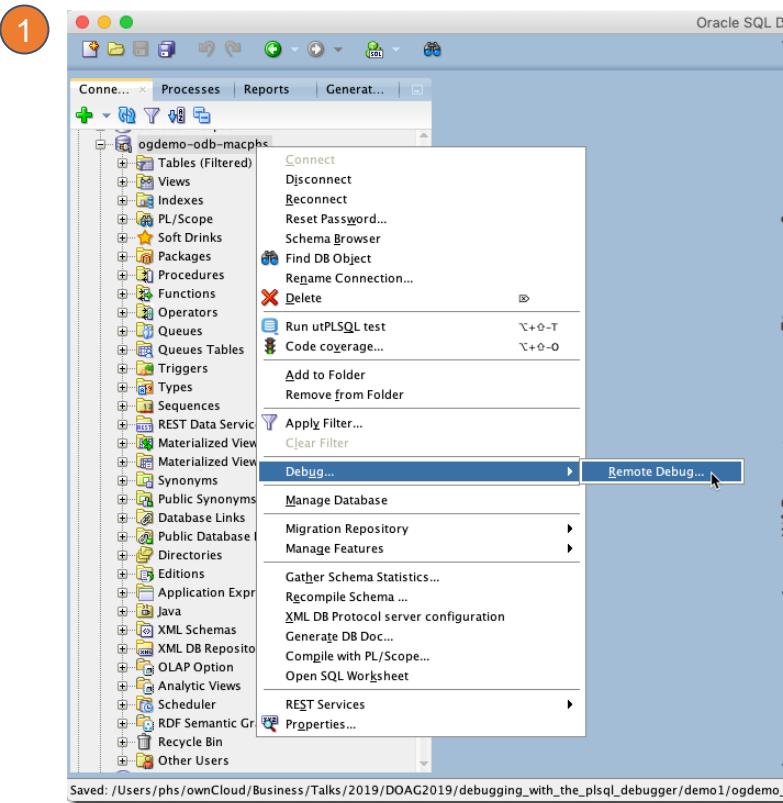
Two orange thought bubbles provide explanatory text:

- The top bubble contains the code `sys.owa_util.get_cgi_env ('REMOTE_ADDR')`.
- The bottom bubble contains the text "parsing schema must be set (wwv_flow_security_pkg)".

Connect From Browser to SQL Developer



Listen for Incoming Debug Connections



Connect to Debugger from APEX App

The screenshot shows the Oracle Application Express (APEX) App Builder interface. The top navigation bar includes links for ORACLE, App Builder, SQL Workshop, Team Development, and App Gallery, along with various search and user icons.

The main workspace displays an application named "Application 100 \ Page Designer". On the left, a sidebar menu lists "Computations", "Validations", "Processes", "Processing" (with "start_debug" selected), "Branches", "After Processing", and "Go To Page 2".

The central area shows a "Form on DEPT (2)" region. The "PL/SQL Code" section contains the following PL/SQL block:

```
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;
```

A red oval highlights the line `dbms_debug_jdwp.connect_tcp('sqldev', 4000);`.

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



MR. ROBOT

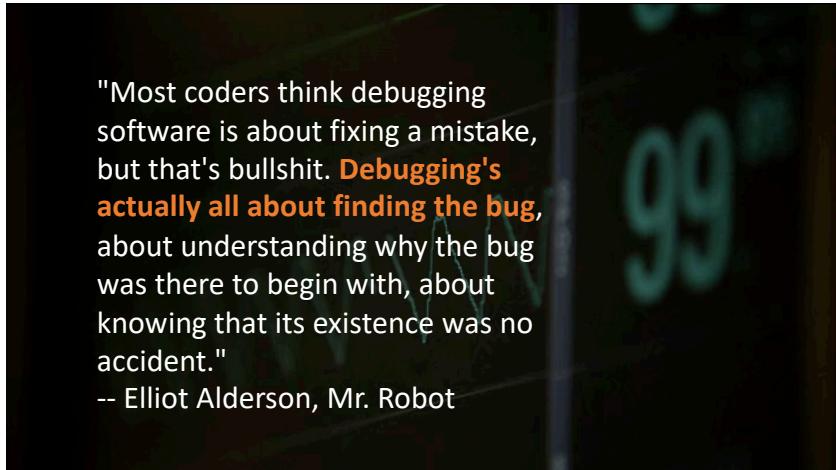


CREATED BY

SAM ESMAIL

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



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