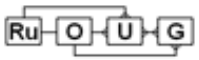
My Favorite Built-in Packages

and a few

PL/SQL Tips & Tricks

presentation for:
OUG Harmony 2011



Daniel A. Morgan | **damorgan11g@gmail.com** | **www.morganslibrary.org**

Disclaimer

This room is an unsafe harbour

No one from Oracle has previewed this presentation

No one from Oracle knows what I'm going to say

No one from Oracle knows what I'm going to demo

No one from Oracle has supplied any of my materials

This presentation will include live in SQL*Plus demos because the technology is currently available and works very very well

You may rely upon this presentation to make decisions for your enterprise

This disclaimer has not been approved by Oracle Legal

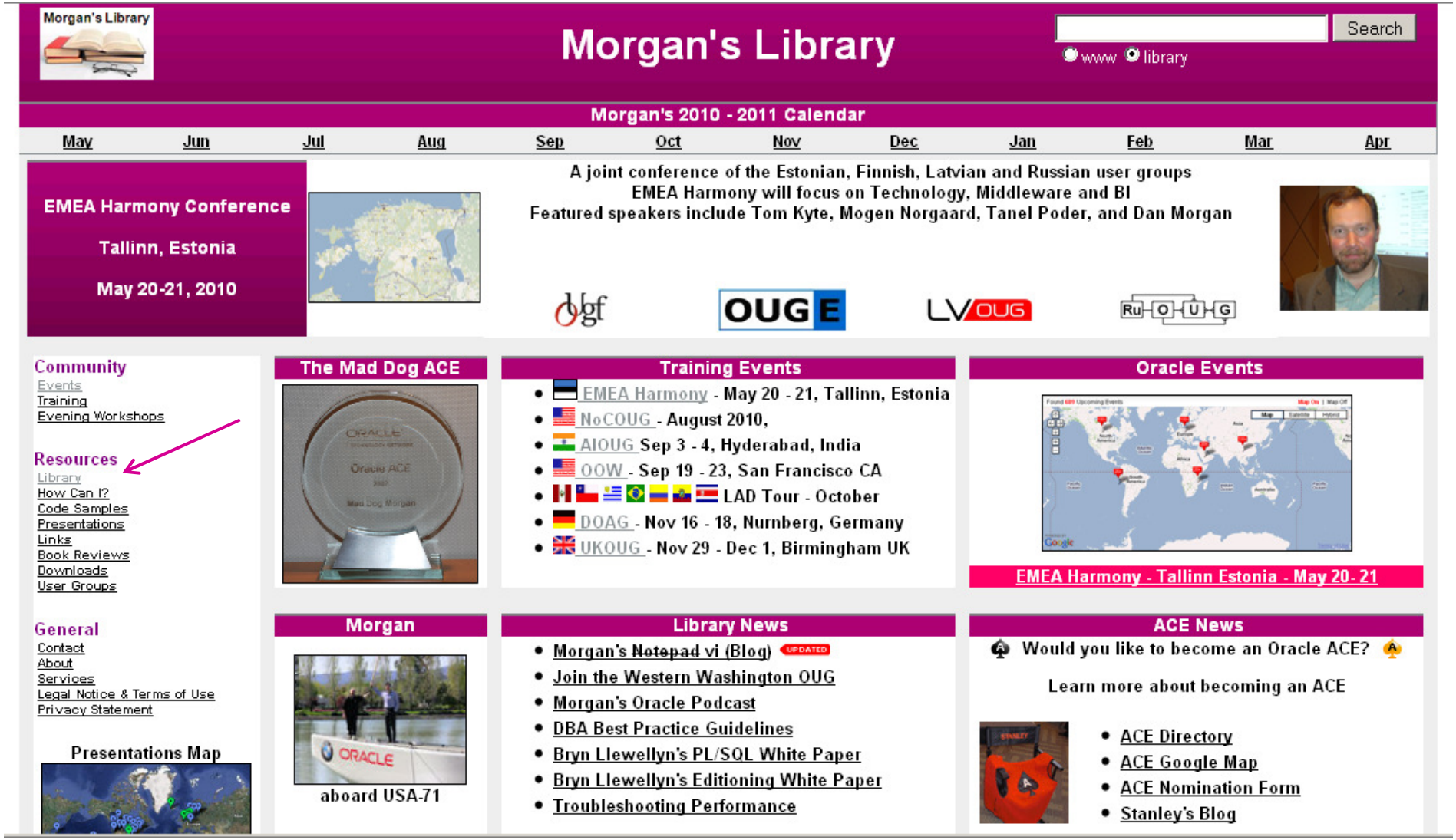
Daniel A. Morgan

- Oracle ACE Director
- University of Washington Oracle Instructor for 10 years
- The Morgan of Morgan's Library on the web
- Board Member: Western Washington OUG
- Member UKOUG
- Conference Speaker
  - OpenWorld, Collaborate, Kaleidoscope, Brazil, Bulgaria, Canada, Chile, Costa Rica, Denmark, Estonia, Finland, Germany, Japan, New Zealand, Norway, Peru, Sweden, U.K., U.S., Uruguay
- 10g & 11g Beta Tester

cd $MORGAN_HOME

cd $MORGAN_HOME

Morgan's Library: www.morganslibrary.org



Morgan's Library Demos

DDL Statements	11gR2	20-Dec-2009	-
Deadlocks	11gR2	09-Sep-2010	UPDATED
DECODE Function	11gR2	21-Apr-2009	-
Deferrable Constraints	11gR2	09-Sep-2009	-
Delete Statement	11gR2	10-Sep-2009	-
Descending Indexes	11gR2	22-Aug-2010	UPDATED
DICOM	11gR1	17-Jun-2010	-
Dimensions	11gR2	10-Sep-2009	-
Directories	11gR2	01-Nov-2009	-
Disassociate Statistics	11gR2	10-Sep-2009	-
DIUTIL	11gR2	10-Sep-2009	-
DML Statements	11gR2	10-Sep-2009	-
Dumping Oracle	11gR2	20-Jul-2010	-
Dynamic Performance Views	11gR2	15-Jun-2010	-
E-Business Suite	11.5.10	29-Sep-2007	-
Edition Based Redefinition	11gR2	18-Sep-2010	UPDATED
Editioning Demo 1: Editions	11gR2	15-Apr-2010	-
Editioning Demo 2: Editioning Views	11gR2	21-Mar-2010	-
Editioning Demo 3: Crossedition Triggers	11gR2	02-Mar-2010	-
Editioning Demo 4: Online Table Update	11gR2	22-Mar-2010	-
Editioning Demo 5: Invisible Indexes	11gR2	09-May-2010	-
Editioning Demo 6: Conversion Automation	11gR2	22-Sep-2010	NEW
Editioning Views	11gR2	24-Feb-2010	-
Editions	11gR2	12-Mar-2010	-
Encrypted Tablespaces	11gR2	28-Sep-2009	-
Environment Variables	11gR2	11-Sep-2009	-
Errors	11gR2	11-Sep-2009	-
Events	11gR2	28-Aug-2010	UPDATED
Exadata	11gR2	24-Dec-2009	-
Exception Handling	11gR2	16-Mar-2010	-
Excluded Nodes	11gR2	27-Sep-2008	-
Exists	11gR2	11-Sep-2009	-
Explain Plan	11gR2	21-Jan-2010	-

Basic Principles

- Code as though every line will run on RAC
- Code as though performance issues are everywhere
- Code as though you will have production issues
- Process sets not single rows
- Do it in memory not on disk

Topics

- Work with Records
- Multidimensional Collections
- PL/SQL Warnings
- Java Functions
- Columns in Weakly Typed REF CURSORS
- Write to V$SESSION
- Audit Specific Column Updates

PL/SQL Code

Work with Records

- Not just BULK COLLECT and FORALL
- You can use BULK COLLECT and FORALL with Native Dynamic SQL
- You can use FORALL with UPDATE and DELETE
- Use the SAVE EXCEPTIONS clause

```
CREATE TABLE t AS
 SELECT table_name, tablespace_name
 FROM all_tables;

 SELECT COUNT(*)
 FROM t;

 DECLARE
  trec  t%ROWTYPE;
 BEGIN
   trec.table_name := 'NEW';
   trec.tablespace_name := 'NEW_TBSP';

   INSERT INTO t
   VALUES trec;

   COMMIT;
 END;
 /

 SELECT COUNT(*) FROM t;
```

```
DECLARE
 trec  t%ROWTYPE;
BEGIN
  trec.table_name := 'DUAL';
  trec.tablespace_name := 'NEW_TBSP';

  UPDATE t
  SET ROW = trec
  WHERE table_name = 'DUAL';

  COMMIT;
END;
/
```

Multi-dimensional Collections

- Collections have one dimension, but you can model a multidimensional collection with a collection whose elements are collections

```
CREATE OR REPLACE TYPE uw_varray AS VARRAY(5) OF NUMBER;
/

CREATE TABLE mdc_tab (
rid         NUMBER,
demo_varray uw_varray);

desc mdc_tab

INSERT INTO mdc_tab VALUES (1, uw_varray(1,2,3,4,5));
INSERT INTO mdc_tab VALUES (2, uw_varray(10,20,30,40,50));
INSERT INTO mdc_tab VALUES (3, uw_varray(100,200,300,400,500));

SELECT * FROM mdc_tab;
```

Multi-dimensional Collections

```
set serveroutput on

DECLARE
 TYPE rid_t IS TABLE OF mdc_tab.rid%TYPE INDEX BY BINARY_INTEGER;
 rid_a   rid_t;

 TYPE uwv_t IS TABLE OF mdc_tab.demo_varray%TYPE INDEX BY BINARY_INTEGER;
 uwv_a   uwv_t;

 exStr   CLOB := 'SELECT rid, demo_varray FROM mdc_tab';
BEGIN
  EXECUTE IMMEDIATE exStr BULK COLLECT INTO rid_a, uwv_a;

  -- execute each row of the array
  FOR i IN rid_a.FIRST .. rid_a.LAST LOOP
    dbms_output.put_line('Accessing array row: ' || TO_CHAR(i));
    FOR j IN uwv_a(i).FIRST .. uwv_a(i).LAST LOOP
      dbms_output.put_line(uwv_a(i)(j));
    END LOOP;
  END LOOP;
END;
/
```

PL/SQL Warnings

- Introduced in Database 10gR1
  - Informational
  - Performance
  - Severe
- ALTER SYSTEM
- ALTER SESSION
  - glogin.sql

```
set pagesize 45
set linesize 121
set long 1000000
col name format a30
col value format a30
col object_name format a30
col segment_name format a30
col file_name format a60
col data_type format a20
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';
ALTER SESSION SET PLSQL_WARNINGS='ENABLE:ALL';
```

PL/SQL Warnings

```
CREATE OR REPLACE PROCEDURE plw06002 AS
 x NUMBER(5) := 10;
BEGIN
  WHILE x < 100 LOOP
    x := 10;
  END LOOP;
END plw06002;
/
```

```
CREATE OR REPLACE PROCEDURE plw06002 wrapped
 a000000
 b2
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 abcd
 7
 6d a2
 ASXzD4E60Xyv09790pDE+Doltlcwg5nnm7+fMr2ywFwWhT54fIJ8fAmldIsJ42mlmYEywLIl
 fD3DS8JcfCF8S4vAwDL+0oYJaef+pdJSROVL+73Vzs74W6ZElKnhcAI8Z0oKT5lUaOyZ256l
 vVv9ftempqqKBBE=
 /
```

Java Functions

- There are things you can do in Java ... you can not do in pure PL/SQL

```
CREATE OR REPLACE FUNCTION get_java_system_property(prop IN VARCHAR2) RETURN VARCHAR2
AUTHID DEFINER IS
LANGUAGE JAVA
  name 'java.lang.System.getProperty(java.lang.String) return java.lang.String';
/

SELECT object_type
FROM user_objects
WHERE object_name = 'GET_JAVA_SYSTEM_PROPERTY';

CREATE OR REPLACE VIEW v$oracle_home AS
SELECT get_java_system_property('user.dir') AS oracle_home
FROM dual;

SELECT * FROM v$oracle_home;
```

Columns in Weakly Typed REF CURSORS

- How Can I?
  - Read a list of operating system files into a PL/SQL array (without using C or Java)
  - Identify the columns and data types in a weakly typed ref cursor (How Can I #3)
  - Write a CLOB to a file
  - Track the parts of my application that are in use
  - Create a data type that will only hold the values 0 and 1
  - Audit Column Updates
  - Only report the rows from a query if the number of rows exceeds a limiting value
  - Work effectively with the deprecated LONG data type
  - Control services and scheduled jobs at startup
  - Perform string aggregation

Write to V$SESSION

- How Can I?
  - Read a list of operating system files into a PL/SQL array (without using C or Java)
  - Identify the columns and data types in a weakly typed ref cursor
  - Write a CLOB to a file
  - Track the parts of an application that are in use (How Can I #5)
  - Create a data type that will only hold the values 0 and 1
  - Audit Column Updates
  - Only report the rows from a query if the number of rows exceeds a limiting value
  - Work effectively with the deprecated LONG data type
  - Control services and scheduled jobs at startup
  - Perform string aggregation

Audit Column Updates

- How Can I?
  - Read a list of operating system files into a PL/SQL array (without using C or Java)
  - Identify the columns and data types in a weakly typed ref cursor
  - Write a CLOB to a file
  - Track the parts of my application that are in use
  - Create a data type that will only hold the values 0 and 1
  - Audit Specific Column Updates (How Can I #8)
  - Only report the rows from a query if the number of rows exceeds a limiting value
  - Work effectively with the deprecated LONG data type
  - Control services and scheduled jobs at startup
  - Perform string aggregation

Built-in Packages

Topics

- DBMS_ASSERT
- DBMS_DB_VERSION
- DBMS_METADATA
- DBMS_RESULT_CACHE
- OWA_UTIL

DBMS_ASSERT Stops SQL Injection

DBMS_ASSERT

- Provides functions which assert various properties of the input value. If the condition which determines the property asserted in a function is not met then a value error is raised. Otherwise the input value is returned via return value. Most functions return the value unchanged, however, several functions modify the value.

This is not English as anyone I know speaks it

DBMS_ASSERT

- Utilities to prevent SQL Injection in dynamic SQL
  - ENQUOTE_LITERAL
  - ENQUOTE_NAME
  - QUALIFIED_SQL_NAME
  - NOOP
  - SCHEMA_NAME
  - SIMPLE_SQL_NAME
  - SQL_OBJECT_NAME

DBMS_DB_VERSION

- Produce a single code base that performs version dependent compilation

```
set serveroutput on

BEGIN
  $IF dbms_db_version.ver_le_10 $THEN
    dbms_output.put_line('version 10 and earlier
code');
  $ELSIF dbms_db_version.ver_le_11 $THEN
    dbms_output.put_line('version 11 code');
  $ELSE
    dbms_output.put_line('version 12 and later
code');
  $END  -- note that there is no semi-colon
END;
/
```

DBMS_METADATA

- Utilities for working with object metadata
  - GET_DEPENDENT_DDL
  - GET_GRANTED_DDL

DBMS_RESULT_CACHE

- Utilities for managing the Result Cache
  - BYPASS
  - DELETE_DEPENDENCY
  - FLUSH
  - INVALIDATE
  - INVALIDATE_OBJECT
  - MEMORY_REPORT
  - STATUS

OWA_UTIL

- Utilities for the now non-existent Oracle Web Application Server

```
set serveroutput on

CREATE OR REPLACE PROCEDURE child AUTHID DEFINER IS
oname all_objects.owner%TYPE;
 pname all_objects.object_name%TYPE;
 lnumb all_source.line%TYPE;
 callr all_objects.object_type%TYPE;
 PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  owa_util.who_called_me(oname, pname, lnumb, callr);
  dbms_output.put_line(oname);
  dbms_output.put_line(pname);
  dbms_output.put_line(lnumb);
  dbms_output.put_line(callr);
END;
/
```

Questions

ERROR at line 1:

ORA-00028: your session has been killed

All demos at morganslibrary.org

- **Click on: Library**
- **Click on: How Can I?**

Thank you