



# PL/SQL Packaged Utilities

and External Tables

Mike Hillanbrand

[MJHillanbrand@pplweb.com](mailto:MJHillanbrand@pplweb.com)

# Extract Table Data

- Found on the internet in pieces over 5 years ago
- Produce Multiple format files from any table or view
  - .CSV
  - Any delimiter character on the keyboard
  - Fixed width columns from a view
  - Ability to use a where clause
  - With or without Headings
- Written to an Oracle Directory

# Extract Table Data

- What is needed to use the code:
  - An NFS or Samba mount point
    - Requires an RFW to Linux support
      - provide the source mount point, server, target mount point and server
      - If Samba you may need to supply a user account and password (non-human)
  - An Oracle directory
    - Create directory out\_dir as 'u1/samba/share/my\_project/out\_dir'
      - Usually requires DBA privileges
    - Permission to use the directory by the production account
      - Grant read, write on directory out\_dir to cl2pra;
      - Usually requires DBA privileges

# Extract Table Data the input elements

- v\_table\_name      IN VARCHAR2,
- v\_directory      IN VARCHAR2,
- v\_file            IN VARCHAR2,
- v\_separator      IN VARCHAR2,
- v\_where\_clause   IN VARCHAR2,
- v\_headings       IN VARCHAR2,
- v\_query           IN VARCHAR2

# Extract Table Data Let's Test

- With heading, column separated
  - `exec extract_table_data.p_dump_table_to_csv ('EMP' , 'OUT_DIR', 'EMP.csv', ',', null, null, null );`
- With custom heading, column separated
  - `exec extract_table_data.p_dump_table_to_csv ('EMP' , 'OUT_DIR', 'EMP_custom.csv', ',', null, 'Emp#,Name,Position,Manager,HireDate,Salary,Commission,Department', null );`

# Extract Table Data - Let's Test

- With custom heading and a where clause, column separated  
exec extract\_table\_data.p\_dump\_table\_to\_csv  
( 'EMP' , 'OUT\_DIR', 'EMP\_analyst.csv', ',', 'where job= "ANALYST" ',  
'Emp#,Name,Position,Manager,HireDate,Salary,Commission,Depart  
ment', null );
- Without heading, column separated  
exec extract\_table\_data.p\_dump\_table\_to\_csv  
( 'EMP' , 'OUT\_DIR', 'EMP\_nohead.csv', ',', null, 'NONE', null );
- Without heading, fixed width, no separator  
exec extract\_table\_data.p\_dump\_table\_to\_csv  
( 'EMP\_fixed\_VW' , 'OUT\_DIR', 'EMP.dat', null, null, 'NONE', null  
);

# Extract Table Data Summary

- Fast unload of delimited or fixed width data (1 million rows in less than 2 minutes)
- Choice of delimiters
- Ability to change headers
- Works on tables or views

# E-Mail via UTL.SMTP

- Eliminate the use of the spooling from SQL\*Plus and mailing output with the korn shell mailx program
- Used with the Extract Table Data routine presents a better face to the customer.
  - Mail attachments from an Oracle Directory
- Maintain PDL's for batch jobs in a table instead of environment files that need to be promoted
  - Grant insert, update access to someone to maintain



# E-Mail via UTL.SMTP Permissions

- Execute on SYS.UTL.SMTP
- Execute on SYS.UTL.TCP
- ACL's

```
BEGIN
DBMS_NETWORK_ACL_ADMIN.CREATE_ACL (
acl => 'smtp-gate-permissions.xml',
description => 'Permissions for smtp gate',
principal => 'SCOTT',
is_grant => TRUE,
privilege => 'connect');
COMMIT;
END;
/
BEGIN
DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL (
acl => 'smtp-gate-permissions.xml',
host => 'mailhub.ppl.com', -- may require a * instead on linux/unix
lower_port => 25,
upper_port => null);
COMMIT;
END;
/
```

# E-Mail via UTL.SMTP the input elements

- p\_email\_to IN VARCHAR2,
- p\_email\_to\_recipient\_parm\_name IN VARCHAR2,
- p\_input\_attachment\_file\_name IN VARCHAR2,
- p\_email\_to\_group\_description IN VARCHAR2,
- p\_email\_subject\_text IN VARCHAR2,
- p\_email\_body\_text IN VARCHAR2,
- p\_email\_attachment\_name IN VARCHAR2,
- p\_attachment\_directory IN VARCHAR2

# E-Mail via UTL.SMTP

```
DECLARE
l_exit_code number;
begin
    email_utils.send_email
    (null,
    'MAIL_RECIPIENT',
    'EMP.csv',
    'HR Salary Review',
    'Test of mail with an attachment',
    'This is the file generated in the extract test',
    'EMP.csv',
    'OUT_DIR',
    l_exit_code);
END;
/
```

# E-Mail via UTL.SMTP

- Util\_Parameter table
  - parameter\_name                    VARCHAR2(30),
  - parameter\_value                    VARCHAR2(200),
  - fl\_test                                CHAR(1),
  - fl\_production                        CHAR(1),
  - fl\_dev                                 CHAR(1)

# E-Mail via UTL.SMTP

- Parameter table mandatory entries
  - PROD\_MAIL\_HOST, mailhub.ppl.com, N, Y, N
  - PROD\_MSG\_FROM,  
noreply.PROD.oracle@pplweb.com, N, Y, N
  - A PDL entry
    - MAIL\_RECIPIENT, mjhillanbrand@pplweb.com, Y, Y, Y
    - MAIL\_RECIPIENT, ddkline@pplweb.com, Y, Y, Y
- Let's test

# E-Mail Summary

- Slow on a laptop database☺
- Ability to use different PDL's from the same table driven model in DEV/TEST/PROD
- Mail any file in an Oracle Directory
- Customize your e-mail based upon programmatic conditions.
- BONUS: Up to 16 attachments

# Data Maintenance

- Delete from a table based on a date
- Delete from a table based on a where clause
- Truncate a table
- Drop date ranged partitions (update indexes)
- Table Driven
- Ability to mark as ineligible for maintenance
- Easy to change the retention period

# Data Maintenance

- Log Table of activity (can set to self-maintain)
- Log Table records user and workstation
- Only 3 jobs needed to maintain the entire database
- Grant update, insert to Developers to the driver table or insert new rules via a controlled environment (PVCS)
- Package owner needs access to maintain tables in the rest of the database (delete any table, truncate any table)



# Data Maintenance Supporting Tables

- Maintain\_Data\_Rules
- Maintain\_Data\_Rules\_Hist
- Data\_Maintenance\_Log
- Tmp\_Mdata\_Rules (Global Temp Table)
- Tmp\_Partition\_Action (Global Temp Table)

# Data Maintenance Driver Table Columns

- owner VARCHAR2(30) NOT NULL,
- table\_name VARCHAR2(30) NOT NULL,
- maintenance\_rule VARCHAR2(20) NOT NULL,
- column\_name VARCHAR2(30),
- date\_type VARCHAR2(12),
- frequency VARCHAR2(20) NOT NULL,
- maintenance\_period NUMBER,
- trim\_method VARCHAR2(20),
- maintenance\_eligible VARCHAR2(1),
- created TIMESTAMP (6) DEFAULT  
SYSTIMESTAMP NOT NULL,
- updated TIMESTAMP (6) DEFAULT  
SYSTIMESTAMP NOT NULL,
- trim\_clause VARCHAR2(400 BYTE),
- tablespace\_name VARCHAR2(30 BYTE)

# Data Maintenance Driver Table Values

- owner any schema you have permissions to act on
- table\_name no validation
- maintenance\_rule TRIM, CREATE
- column\_name date column for comparison - no validation
- date\_type DATE, SAS\_DATE, TIMESTAMP
- frequency DAY, WEEK, MONTH
- maintenance\_period retention period in frequency
- trim\_method RANGEPARTITION, DELETE, TRUNCATE, WHERE, CREATE
- maintenance\_eligible Y/N
- created 26-JUN-2014 11:21:21.641000 AM
- updated 26-JUN-2014 11:21:21.641000 AM
- trim\_clause follows where keyword (not in...)
- tablespace\_name for a create statement - new partitions - no validation

# Data Maintenance Driver Table Sample

• Owner	SCOTT
• table_name	AMRHOURLYDATA
• maintenance_rule	TRIM
• column_name	LOG_DATE
• date_type	DATE
• Frequency	MONTH
• maintenance_period	3
• trim_method	RANGEPARTITION
• maintenance_eligible	Y
• created	26-JUN-2014 11:21:21.641000 AM
• updated	26-JUN-2014 11:21:21.641000 AM
• trim_clause	
• tablespace_name	

# Data Maintenance

- Let's Test

# Data Maintenance Summary

- Any type of table
- Truncate or delete
- Fire statistics (deprecated)
- Truncate partitioned tables
- Table driven
- Delete using a where clause

# External Tables

- Use instead of SQL\*Loader
- No need for staging tables
- With PL/SQL you can perform complex loads
- No Indexes though
- Needs an Oracle Directory with read and write permissions

# External Tables

```
CREATE TABLE salary_update_2014
```

```
( empno      number,  
  ename      varchar2(10),  
  job        varchar2(9),  
  mgr        number,  
  hiredate   date,  
  sal        number(7,2),  
  comm       number(7,2),  
  deptno     number(2,0))
```

```
ORGANIZATION EXTERNAL
```

```
( TYPE ORACLE_LOADER  DEFAULT DIRECTORY IN_DIR  ACCESS  
PARAMETERS
```

```
  (records delimited BY newline
```

```
  fields terminated BY ','
```

```
  missing field VALUES are NULL )
```

```
LOCATION ('EMP_updated.csv') );
```



# External Tables

- Let's Test

```
update emp a
```

```
set sal = (select sal from  
salary_update_2014 where empno  
= a.empno);
```

```
commit;
```

# External Tables Summary

- Instead of SQL\*Loader - no loss in speed
- No need for a staging table
- Complex cursor driven operations permitted
- Simple inserts useful as well
- Needs the input file in an Oracle Directory
- No indexes