

Advanced PLSQL

Lesson 01: Oracle Supplied
packages

Lesson Objectives

- To understand the following topics:
- Oracle Supplied Packages like
 - DBMS_OUTPUT
 - UTL_FILE
 - UTL_MAIL



Oracle supplied packages

- The Oracle-supplied packages:
 - Are provided with the Oracle server
 - Extend the functionality of the database
 - Enable access to certain SQL features that are normally restricted for PL/SQL
- For example, the `DBMS_OUTPUT` package was originally designed to debug PL/SQL programs.

Examples of Some Oracle-Supplied Packages

- Here is an abbreviated list of some Oracle-supplied packages:
 - DBMS_OUTPUT
 - UTL_FILE
 - UTL_MAIL
 - DBMS_ALERT
 - DBMS_LOCK
 - DBMS_SESSION
 - DBMS_APPLICATION_INFO
 - HTP
 - DBMS_SCHEDULER



Copyright © Capgemini 2015. All Rights Reserved. 4

Add the notes here.

1.1: Oracle Supplied Packages:DBMS_OUTPUT DBMS_OUTPUT package

- PL/SQL has no input/output capability
- However, built-in package DBMS_OUTPUT is provided to generate reports
- The procedure PUT_LINE is also provided that places the contents in the buffer

PUT_LINE (VARCHAR2 OR NUMBER OR DATE)



Copyright © Capgemini 2015. All Rights Reserved 5

1.1: Oracle Supplied
Packages:DBMS_OUTPUT
How the DBMS_OUTPUT Package Works

- The DBMS_OUTPUT package enables you to send messages from stored subprograms and triggers.
 - PUT and PUT_LINE place text in the buffer.
 - GET_LINE and GET_LINES read the buffer.
 - Messages are not sent until the sending subprogram or trigger completes.
 - Use SET SERVEROUTPUT ON to display messages in SQL Developer and SQL*Plus.

SET SERVEROUT ON [SIZE n]
EXECUTE proc

Output

PL/SQL

PUT
NEW_LINE
PUT_LINE

GET_LINE
GET_LINES

Buffer

Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved. 6

Add the notes here.

DBMS_OUTPUT package

- Syntax:

```
SQL>SET SERVEROUTPUT ON
DECLARE
  V_Variable VARCHAR2(25) :=' Used for'
  || 'Debugging ';
BEGIN
  DBMS_OUTPUT.PUT_LINE(V_Variable);
END;
```



Copyright © Capgemini 2015. All Rights Reserved

7

The code will be written on SQL prompt

1.1: Oracle Supplied

Packages:DBMS_OUTPUT

Displaying Output

- DBMS_OUTPUT:
 - DBMS_OUTPUT provides a mechanism for displaying information from the PL/SQL program on to your screen (that is your session's output device).
 - The DBMS_OUTPUT package is created when the Oracle database is installed.
 - The "dbmsoutp.sql" script contains the source code for the specification of this package.
 - This script is called by the "catproc.sql" script, which is normally run immediately after database creation.

 Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved. 8

Note:

- The catproc.sql script creates the public synonym DBMS_OUTPUT for the package.
- Instance-wise access to this package is provided on installation, so no additional steps should be necessary in order to use DBMS_OUTPUT.

1.1: Oracle Supplied

Packages:DBMS_OUTPUT

DBMS_OUTPUT Program Names

Table: DBMS_OUTPUT programs

Name	Description	Use in SQL?
DISABLE	Disables output from the package; the DBMS_OUTPUT buffer will not be flushed to the screen.	Yes
ENABLE	Enables output from the package.	Yes
GET_LINE	Gets a single line from the buffer.	Yes
GET_LINES	Gets specified number of lines from the buffer and passes them into a PL/SQL table.	Yes
NEW_LINE	Inserts an end-of-line mark in the buffer.	Yes
PUT	Puts information into the buffer.	Yes
PUT_LINE	Puts information into the buffer and appends an end-of-line marker after that data.	Yes

 Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved. 9

DBMS_OUTPUT Concepts:

- Each user has a DBMS_OUTPUT buffer of up to 1,000,000 bytes in size. You can write information to this buffer by calling the DBMS_OUTPUT.PUT and DBMS_OUTPUT.PUT_LINE programs.
 - If you are using DBMS_OUTPUT from within SQL*Plus, this information will be automatically displayed when your program terminates.
 - You can (optionally) explicitly retrieve information from the buffer with calls to DBMS_OUTPUT.GET and DBMS_OUTPUT.GET_LINE.
- The DBMS_OUTPUT buffer can be set to a size between 2,000 and 1,000,000 bytes with the DBMS_OUTPUT.ENABLE procedure.
 - If you do not enable the package, no information will be displayed or be retrievable from the buffer.
- The buffer stores three different types of data in their internal representations, namely VARCHAR2, NUMBER, and DATE.
 - These types match the overloading available with the PUT and PUT_LINE procedures.
 - Note that DBMS_OUTPUT does not support Boolean data in either its buffer or its overloading of the PUT procedures.

DBMS_OUTPUT - Example

- In this example, the following anonymous PL/SQL block uses DBMS_OUTPUT to display the name and salary of each staff member in department 10:

```
DECLARE
  CURSOR emp_cur IS SELECT staff_name, staff_sal
  FROM staff_master WHERE dept_code = 10
  ORDER BY staff_sal DESC;
BEGIN FOR emp_rec IN emp_cur
  LOOP
    DBMS_OUTPUT.PUT_LINE ('Employee ' ||
      emp_rec.staff_name || ' earns ' ||
      TO_CHAR (emp_rec.staff_sal) || 'rupees');
  END LOOP;
END;
```



Copyright © Capgemini 2015. All Rights Reserved. 10

DBMS_OUTPUT Concepts:

- The program shown in the slide generates the following output when executed in SQL*Plus:
Employee John earns 32000 rupees
Employee Mohan earns 24000 rupees.

1.1: Oracle Supplied

Packages:DBMS_OUTPUT

Writing to DBMS_OUTPUT buffer

- You can write information to the DBMS_OUTPUT buffer with calls to the PUT, NEW_LINE, and PUT_LINE procedures.



Copyright © Capgemini 2015. All Rights Reserved.

11

Writing to DBMS_OUTPUT buffer

- The DBMS_OUTPUT.PUT procedure:
 - The PUT procedure puts information into the buffer, but does not append a newline marker into the buffer.
 - Use PUT if you want to place information in the buffer (usually with more than one call to PUT), but not also automatically issue a newline marker.
 - The specification for PUT is overloaded, so that you can pass data in its native format to the package without having to perform conversions.

```
PROCEDURE DBMS_OUTPUT.PUT (A VARCHAR2);
```



Copyright © Capgemini 2015. All Rights Reserved. 12

Writing to DBMS_OUTPUT buffer: DBMS_OUTPUT.PUT procedure:

Note:

- The specification for PUT is overloaded, so that you can pass data in its native format to the package without having to perform conversions.

```
PROCEDURE DBMS_OUTPUT.PUT (A VARCHAR2);
PROCEDURE DBMS_OUTPUT.PUT (A NUMBER);
PROCEDURE DBMS_OUTPUT.PUT (A DATE);
```

where A is the data being passed.

Example:

- In the following example, three simultaneous calls to PUT, place the employee name, department ID number, and hire date into a single line in the DBMS_OUTPUT buffer:

```
DBMS_OUTPUT.PUT (:employee.lname || ',' || 
:employee.fname);
DBMS_OUTPUT.PUT (:employee.department_id);
DBMS_OUTPUT.PUT (:employee.hiredate);
```

- If you follow these PUT calls with a NEW_LINE call, that information can then be retrieved with a single call to GET_LINE.

Writing to DBMS_OUTPUT buffer

- The DBMS_OUTPUT.NEW_LINE procedure:
 - The NEW_LINE procedure inserts an end-of-line marker in the buffer.
 - Use NEW_LINE after one or more calls to PUT in order to terminate those entries in the buffer with a newline marker.
 - Given below is the specification for NEW_LINE:

```
PROCEDURE DBMS_OUTPUT.NEW_LINE;
```



Writing to DBMS_OUTPUT buffer

- The DBMS_OUTPUT.PUT_LINE procedure:
 - The PUT_LINE procedure puts information into the buffer, and then appends a newline marker into the buffer.
 - The specification for PUT_LINE is overloaded, so that you can pass data in its native format to the package without having to perform conversions.

```
PROCEDURE DBMS_OUTPUT.PUT_LINE (A VARCHAR2);
```



Copyright © Capgemini 2015. All Rights Reserved 14

Writing to DBMS_OUTPUT buffer: DBMS_OUTPUT.PUT_LINE procedure:

Note:

- The specification for PUT_LINE is overloaded, so that you can pass data in its native format to the package without having to perform conversions:

```
PROCEDURE DBMS_OUTPUT.PUT_LINE (A  
VARCHAR2);  
PROCEDURE DBMS_OUTPUT.PUT_LINE (A NUMBER);  
PROCEDURE DBMS_OUTPUT.PUT_LINE (A DATE);
```

where A is the data being passed

- The PUT_LINE procedure is the one most commonly used in SQL*Plus to debug PL/SQL programs.
- When you use PUT_LINE in these situations, you do not need to call GET_LINE to extract the information from the buffer. Instead, SQL*Plus will automatically dump out the DBMS_OUTPUT buffer when your PL/SQL block finishes executing. (You will not see any output until the program ends.)

contd.

1.1: Oracle Supplied Packages:DBMS_OUTPUT

Retrieving Data from DBMS_OUTPUT buffer

- You can retrieve information from the DBMS_OUTPUT buffer with call to the GET_LINE procedure.
- The DBMS_OUTPUT.GET_LINE procedure:
- The GET_LINE procedure retrieves one line of information from the buffer.
 - Given below is the specification for the procedure:

```
PROCEDURE DBMS_OUTPUT.GET_LINE (line OUT VARCHAR2,
                                status OUT INTEGER);
```

 Capgemini

Copyright © Capgemini 2015. All Rights Reserved. 15

Retrieving Data from the DBMS_OUTPUT buffer:

- You can use the GET_LINE and GET_LINES procedures to extract information from the DBMS_OUTPUT buffer.
- If you are using DBMS_OUTPUT from within SQL*Plus, however, you will never need to call either of these procedures. Instead, SQL*Plus will automatically extract the information and display it on the screen for you.

The DBMS_OUTPUT.GET_LINE procedure:

- The GET_LINE procedure retrieves one line of information from the buffer.
- Given below is the specification for the procedure:

```
PROCEDURE DBMS_OUTPUT.GET_LINE
    (line OUT VARCHAR2,
     status OUT INTEGER);
```

- The parameters are summarized as shown below:

Parameter	Description
Line	Retrieved line of text
Status	GET request status

contd.

1.2: Oracle Supplied

Packages: UTL_FILE Using the UTL_FILE Package

- The UTL_FILE package extends PL/SQL programs to read and write operating system text files:
 - Provides a restricted version of operating system stream file I/O for text files
 - Can access files in operating system directories defined by a CREATE DIRECTORY statement

CREATE DIRECTORY reports_dir AS '/home/oracle/labs/plpu/reports'

EXEC proc

PL/SQL

UTL_FILE

Operating system file

Capgemini

Copyright © Capgemini 2015. All Rights Reserved. 16

Add the notes here.

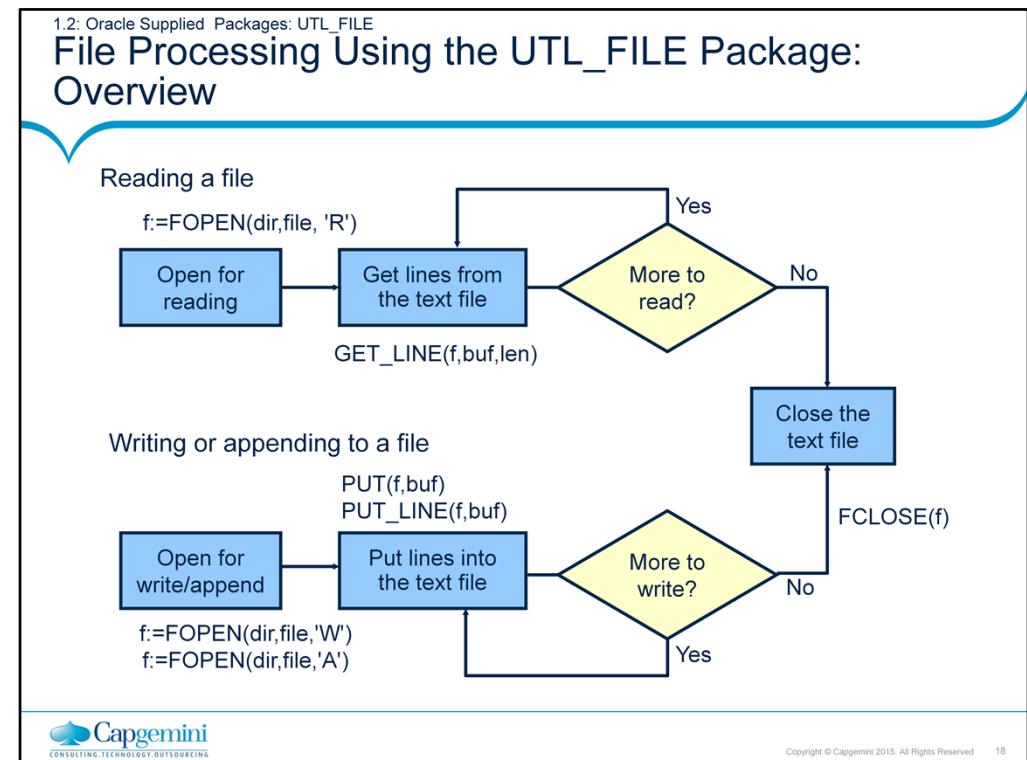
1.2: Oracle Supplied
Packages: UTL_FILE
Some of the UTL_FILE Procedures and Functions

Subprogram	Description
ISOPEN function	Determines if a file handle refers to an open file
FOPEN function	Opens a file for input or output
FCLOSE function	Closes all open file handles
FCOPY procedure	Copies a contiguous portion of a file to a newly created file
FGETATTR procedure	Reads and returns the attributes of a disk file
GET_LINE procedure	Reads text from an open file
FREMOVE procedure	Deletes a disk file, if you have sufficient privileges
FRENAME procedure	Renames an existing file to a new name
PUT procedure	Writes a string to a file
PUT_LINE procedure	Writes a line to a file, and so appends an operating system-specific line terminator



Copyright © Capgemini 2015. All Rights Reserved. 17

Add the notes here.



Add the notes here.

Using the Available Declared Exceptions in the UTL_FILE Package

Exception Name	Description
INVALID_PATH	File location invalid
INVALID_MODE	The open_mode parameter in FOPEN is invalid
INVALID_FILEHANDLE	File handle invalid
INVALID_OPERATION	File could not be opened or operated on as requested
READ_ERROR	Operating system error occurred during the read operation
WRITE_ERROR	Operating system error occurred during the write operation
INTERNAL_ERROR	Unspecified PL/SQL error



Copyright © Capgemini 2015. All Rights Reserved. 19

Add the notes here.

1.2: Oracle Supplied

Packages: UTL_FILE FOPEN and IS_OPEN Functions: Example

- This FOPEN function opens a file for input or output.

```
FUNCTION FOPEN (p_location IN VARCHAR2,  
                p_filename IN VARCHAR2,  
                p_open_mode IN VARCHAR2)  
RETURN UTL_FILE.FILE_TYPE;
```

- The IS_OPEN function determines whether a file handle refers to an open file.

```
FUNCTION IS_OPEN (p_file IN FILE_TYPE)  
RETURN BOOLEAN;
```



Copyright © Capgemini 2015. All Rights Reserved 20

1.2: Oracle Supplied
Packages: UTL_FILE
Using UTL_FILE: Example

```
CREATE OR REPLACE PROCEDURE sal_status(
  p_dir IN VARCHAR2, p_filename IN VARCHAR2) IS
  f_file UTL_FILE.FILE_TYPE;
  CURSOR cur_emp IS
    SELECT last_name, salary, department_id
    FROM employees ORDER BY department_id;
  v_newdeptno employees.department_id%TYPE;
  v_olddeptno employees.department_id%TYPE := 0;
BEGIN
  f_file:= UTL_FILE.FOPEN (p_dir, p_filename, 'W');
  UTL_FILE.PUT_LINE(f_file,
    'REPORT: GENERATED ON ' || SYSDATE);
  UTL_FILE.NEW_LINE (f_file);
  ...
END;
```



Copyright © Capgemini 2015. All Rights Reserved 21

1.2: Oracle Supplied

Packages: UTL_FILE

Using UTL_FILE: Example

```
...
FOR emp_rec IN cur_emp LOOP
  IF emp_rec.department_id <> v_olddeptno THEN
    UTL_FILE.PUT_LINE (f_file,
      'DEPARTMENT: ' || emp_rec.department_id);
    UTL_FILE.NEW_LINE (f_file);
  END IF;
  UTL_FILE.PUT_LINE (f_file,
    'EMPLOYEE: ' || emp_rec.last_name ||
      ' earns: ' || emp_rec.salary);
  v_olddeptno := emp_rec.department_id;
  UTL_FILE.NEW_LINE (f_file);
END LOOP;
UTL_FILE.PUT_LINE(f_file,'*** END OF REPORT ***');
UTL_FILE.FCLOSE (f_file);
EXCEPTION
  WHEN UTL_FILE.INVALID_FILEHANDLE THEN
    RAISE_APPLICATION_ERROR(-20001,'Invalid File.');
  WHEN UTL_FILE.WRITE_ERROR THEN
    RAISE_APPLICATION_ERROR (-20002, 'Unable to write to file');
END sal_status;/
```

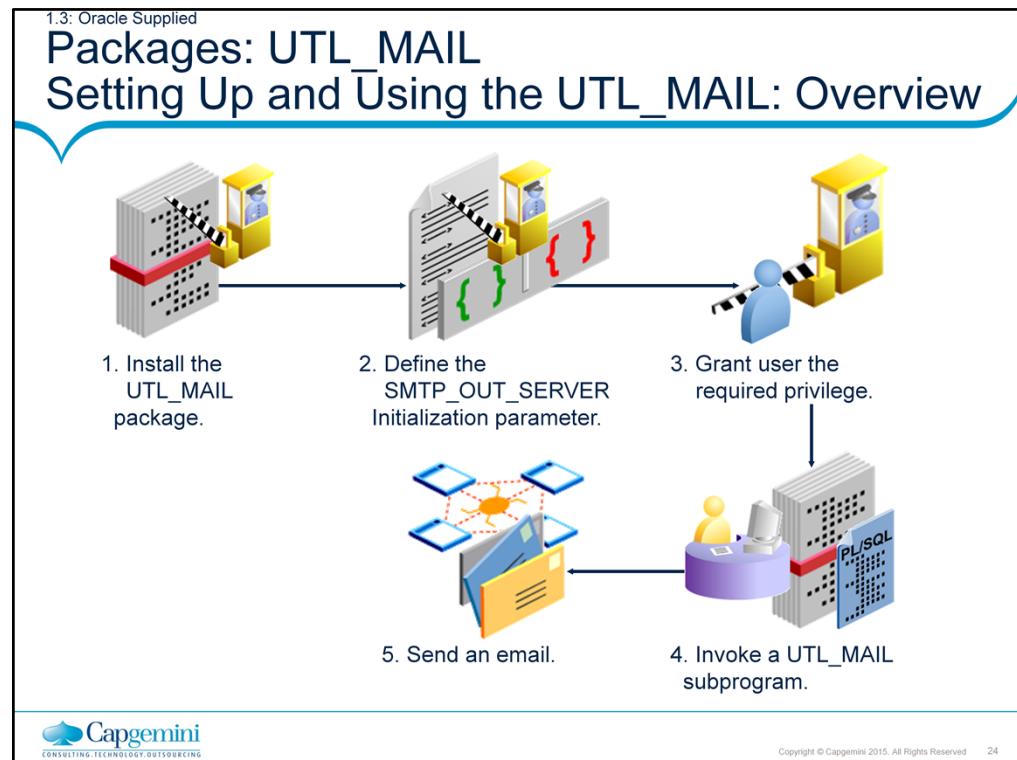


Copyright © Capgemini 2015. All Rights Reserved 22

1.3: Oracle Supplied
Packages: UTL_MAIL
What Is the UTL_MAIL Package?

- A utility for managing email
- Requires the setting of the SMTP_OUT_SERVER database initialization parameter
- Provides the following procedures:
 - SEND for messages without attachments
 - SEND_ATTACH_RAW for messages with binary attachments
 - SEND_ATTACH_VARCHAR2 for messages with text attachments





1.3: Oracle Supplied

Packages: UTL_MAIL

Summary of UTL_MAIL Subprograms

Subprogram	Description
SEND procedure	Packages an email message, locates SMTP information, and delivers the message to the SMTP server for forwarding to the recipients
SEND_ATTACH_RAW Procedure	Represents the SEND procedure overloaded for RAW attachments
SEND_ATTACH_VARCHAR2 Procedure	Represents the SEND procedure overloaded for VARCHAR2 attachments



Copyright © Capgemini 2015. All Rights Reserved 25

1.3: Oracle Supplied

Packages: UTL_MAIL

Installing and Using UTL_MAIL

- As SYSDBA, using SQL Developer or SQL*Plus:

- Install the UTL_MAIL package

```
@?/rdbms/admin/utlmail.sql  
@?/rdbms/admin/prvtmail.plb
```

- Set the SMTP_OUT_SERVER

```
ALTER SYSTEM SET SMTP_OUT_SERVER='smtp.server.com'  
SCOPE=SPFILE
```

- As a developer, invoke a UTL_MAIL procedure:

```
BEGIN  
  UTL_MAIL.SEND('otn@oracle.com','user@oracle.com',  
    message => 'For latest downloads visit OTN',  
    subject => 'OTN Newsletter');  
END;
```



Copyright © Capgemini 2015. All Rights Reserved 26

1.3: Oracle Supplied

Packages: UTL_MAIL

The SEND Procedure Syntax

- Packages an email message into the appropriate format, locates SMTP information, and delivers the message to the SMTP server for forwarding to the recipients



Copyright © Capgemini 2015. All Rights Reserved 27

1.3: Oracle Supplied
Packages: UTL_MAIL
The SEND Procedure Syntax

```
UTL_MAIL.SEND (
    sender      IN      VARCHAR2 CHARACTER SET ANY_CS,
    recipients  IN      VARCHAR2 CHARACTER SET ANY_CS,
    cc          IN      VARCHAR2 CHARACTER SET ANY_CS
                      DEFAULT NULL,
    bcc         IN      VARCHAR2 CHARACTER SET ANY_CS
                      DEFAULT NULL,
    subject     IN      VARCHAR2 CHARACTER SET ANY_CS
                      DEFAULT NULL,
    message     IN      VARCHAR2 CHARACTER SET ANY_CS,
    mime_type   IN      VARCHAR2
                      DEFAULT 'text/plain; charset=us-
                        ascii',
    priority    IN      PLS_INTEGER DEFAULT NULL);
```



Copyright © Capgemini 2015. All Rights Reserved 28

1.3: Oracle Supplied

Packages: UTL_MAIL The SEND_ATTACH_RAW Procedure

- This procedure is the SEND procedure overloaded for RAW attachments.

```
UTL_MAIL.SEND_ATTACH_RAW (
  sender      IN  VARCHAR2 CHARACTER SET ANY_CS,
  recipients   IN  VARCHAR2 CHARACTER SET ANY_CS,
  cc          IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
  bcc         IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
  subject      IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
  message      IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
  mime_type    IN  VARCHAR2 DEFAULT CHARACTER SET ANY_CS
                  DEFAULT 'text/plain; charset=us-ascii',
  priority     IN  PLS_INTEGER DEFAULT 3,
  attachment   IN  RAW,
  att_inline   IN  BOOLEAN DEFAULT TRUE,
  att_mime_type IN  VARCHAR2 CHARACTER SET ANY_CS
                  DEFAULT 'text/plain; charset=us-ascii',
  att_filename IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL);
```



Copyright © Capgemini 2015. All Rights Reserved 29

1.3: Oracle Supplied Packages: UTL_MAIL

Sending Email with a Binary Attachment: Example

```
CREATE OR REPLACE PROCEDURE send_mail_logo IS
BEGIN
  UTL_MAIL.SEND_ATTACH_RAW(
    sender => 'me@oracle.com',
    recipients => 'you@somewhere.net',
    message =>
      '<HTML><BODY>See attachment</BODY></HTML>',
    subject => 'Oracle Logo',
    mime_type => 'text/html'
    attachment => get_image('oracle.gif'),
    att_inline => true,
    att_mime_type => 'image/gif',
    att_filename => 'oralogo.gif');
END;
/
```



Copyright © Capgemini 2015. All Rights Reserved 30

1.3: Oracle Supplied Packages: UTL_MAIL

The SEND_ATTACH_VARCHAR2 Procedure

- This procedure is the SEND procedure overloaded for VARCHAR2 attachments.

```
UTL_MAIL.SEND_ATTACH_VARCHAR2 (
    sender      IN  VARCHAR2 CHARACTER SET ANY_CS,
    recipients   IN  VARCHAR2 CHARACTER SET ANY_CS,
    cc          IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
    bcc         IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
    subject     IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
    message     IN  VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
    mime_type   IN  VARCHAR2 CHARACTER SET ANY_CS
                  DEFAULT 'text/plain; charset=us-ascii',
    priority    IN  PLS_INTEGER DEFAULT 3,
    attachment  IN  VARCHAR2 CHARACTER SET ANY_CS,
    att_inline   IN  BOOLEAN DEFAULT TRUE,
    att_mime_type IN  VARCHAR2 CHARACTER SET ANY_CS
                  DEFAULT 'text/plain; charset=us-ascii',
    att_filename IN  VARCHAR2CHARACTER SET ANY_CS DEFAULT NULL);
```



Copyright © Capgemini 2015. All Rights Reserved 31

1.3: Oracle Supplied Packages: UTL_MAIL

Sending Email with a Text Attachment: Example

```
CREATE OR REPLACE PROCEDURE send_mail_file IS
BEGIN
  UTL_MAIL.SEND_ATTACH_VARCHAR2(
    sender => 'me@oracle.com',
    recipients => 'you@somewhere.net',
    message =>
      '<HTML><BODY>See attachment</BODY></HTML>',
    subject => 'Oracle Notes',
    mime_type => 'text/html',
    attachment => get_file('notes.txt'),
    att_inline => false,
    att_mime_type => 'text/plain',
    att_filename => 'notes.txt');
END;
/
```



Copyright © Capgemini 2015. All Rights Reserved 32

Summary

- In this lesson you have learnt:
 - How the `DBMS_OUTPUT` package works
 - How to use `UTL_FILE` to direct output to operating system files
 - About the main features of `UTL_MAIL`



Copyright © Capgemini 2015. All Rights Reserved 33

Add the notes here.

Review Question

- Which of the following is the most frequently used package in reports
 - DBMS_OUTPUT
 - UTL_FILE
 - SRW
 - DBMS_SQL
- With the UTL_FILE package, PL/SQL programs can read and write operating system text files
 - True/False
- Question 3: _____ can be opened on the server and passed to the client as a unit rather than fetching one row at a time.



Copyright © Capgemini 2015. All Rights Reserved 34

Add the notes here.