Using oracle-supplied Packages in Application Development

Using 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
- HTP
- DBMS_SCHEDULER

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.

DBMS_OUTPUT

- PUT appends text from the procedure to the current line of the line output buffer.
- NEW_LINE places an end-of-line marker in the output buffer.
- PUT_LINE combines the action of PUT and NEW_LINE (to trim leading spaces).
- GET_LINE retrieves the current line from the buffer into a procedure variable.
- GET_LINES retrieves an array of lines into a procedure-array variable.
- ENABLE/DISABLE enables and disables calls to DBMS_OUTPUT procedures.

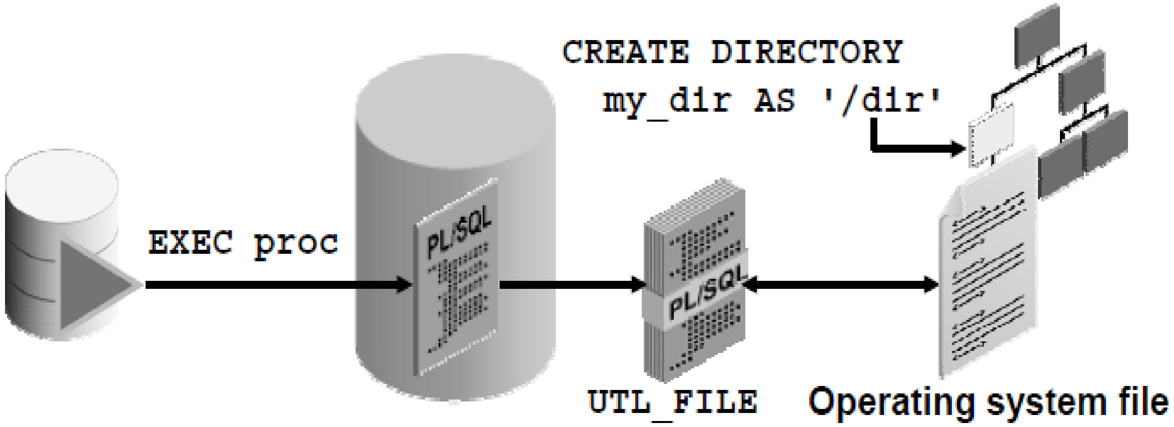
The buffer size
The minimum is 2,000 and the maximum is unlimited. The default is 20,000.
An integer parameter between 2,000 and 1,000,000 in the ENABLE procedure

Actually the DBMS_OUTPUT is not that important for developers
Because they deal with messages using the development applications features
Example: alert windows, console windows

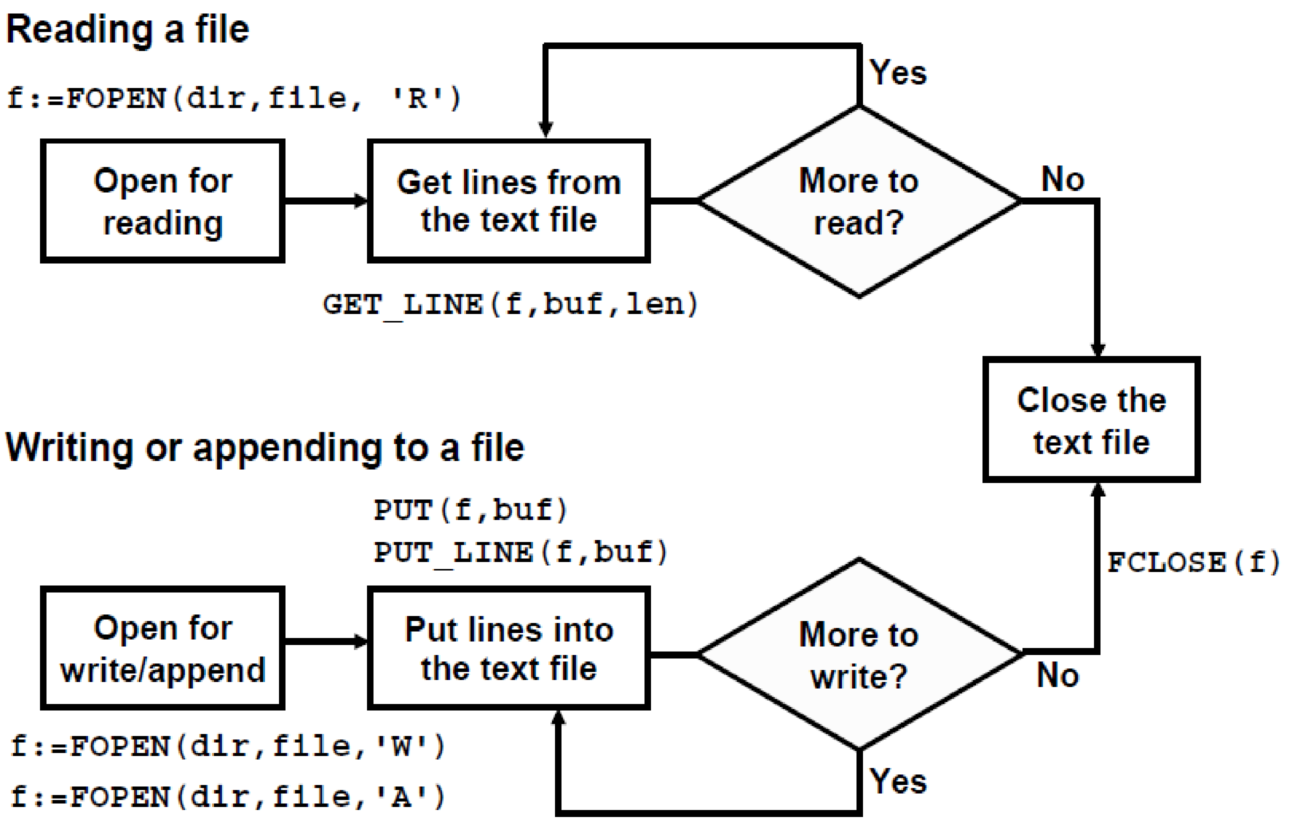
Using the UTL_FILE Package to Interact with Operating System Files

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



File Processing Using the UTL_FILE Package: Overview



Reading a file

f:=FOPEN(dir,file, 'R')

Open for reading

Get lines from the text file

GET_LINE(f,buf,len)

More to read?

Yes

No

Close the text file

FCLOSE(f)

Writing or appending to a file

PUT(f,buf)
PUT_LINE(f,buf)

Open for write/append

Put lines into the text file

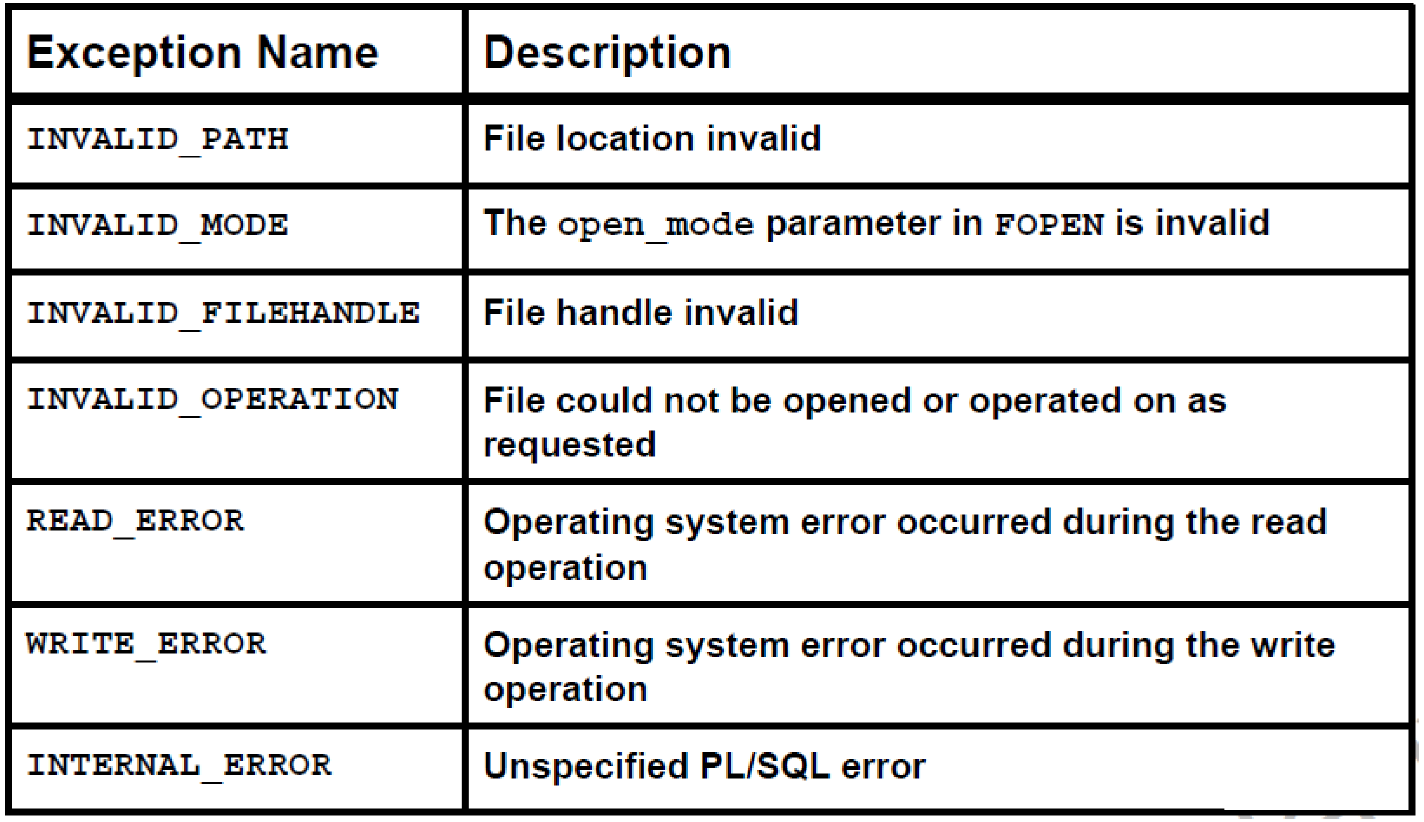
More to write?

No

Yes

f:=FOPEN(dir,file,'W')
f:=FOPEN(dir,file,'A')

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

The FOPEN and IS_OPEN Functions: Example

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

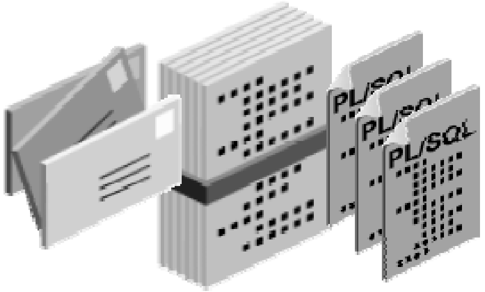
```
FUNCTION FOPEN (p_location  IN VARCHAR2,
                p_filename  IN VARCHAR2,
                p_open_mode IN VARCHAR2)
RETURN UTL_FILE.FILE_TYPE;
```

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

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

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



How UTL_MAIL work ?????

1- THE DBA install UTL_MAIL package

```
@$ORACLE_HOME/rdbms/admin/utlmail.sql
@$ORACLE_HOME/rdbms/admin/prvtmail.plb
```

2- define the SMTP_OUT_SERVER (init.ora)

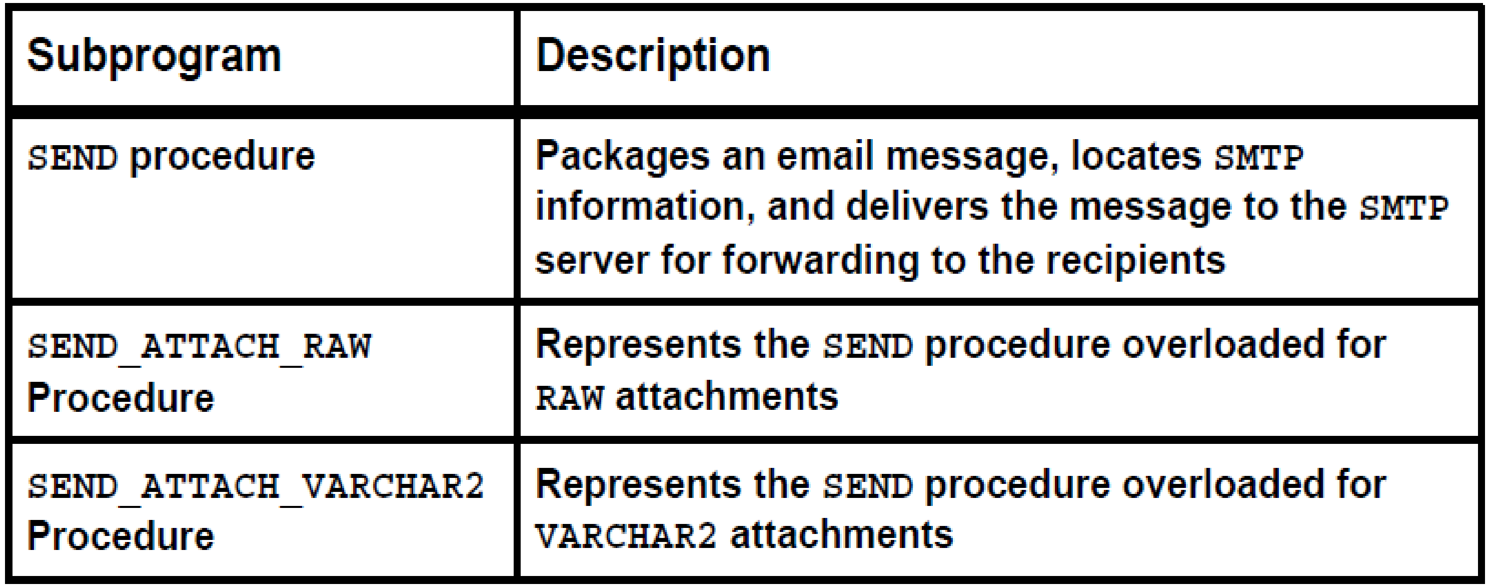
3- the DBA grant Execute on UTL_MAIL to a USER/ Public

4- The DBA add Record in ACL (Access Control List)
Using package called DBMS_NETWORK_ACL_ADMIN

5- now the USER can use UTL_MAIL

Note: No need to know details for steps 1-4

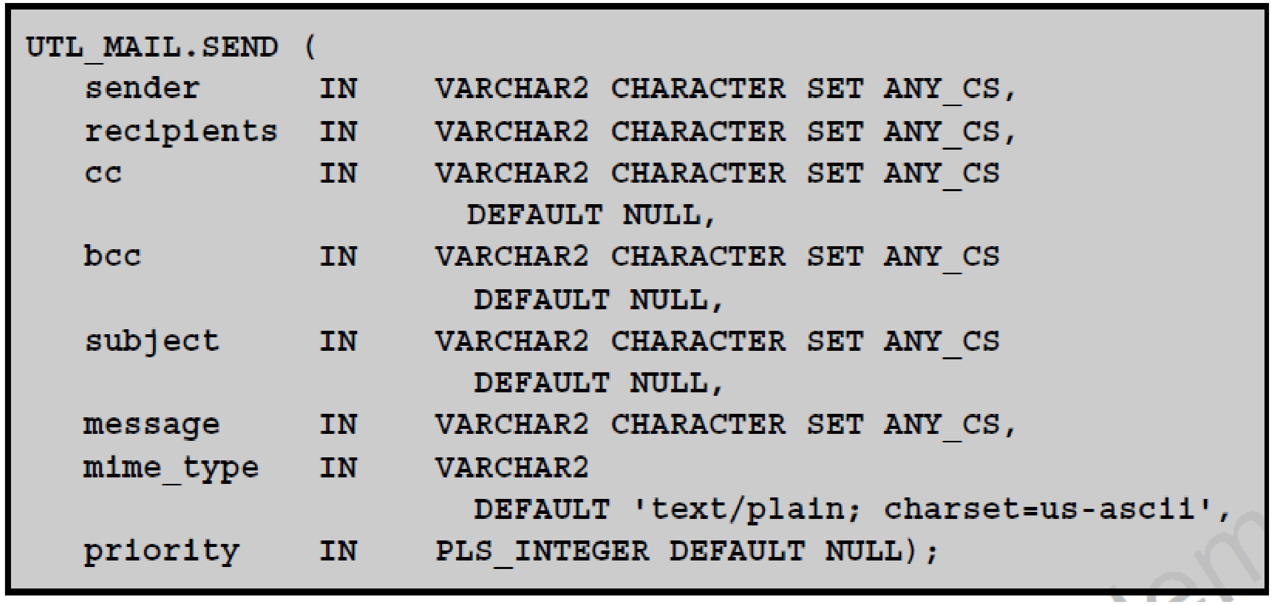
Summary of UTL_MAIL Subprograms



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

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.

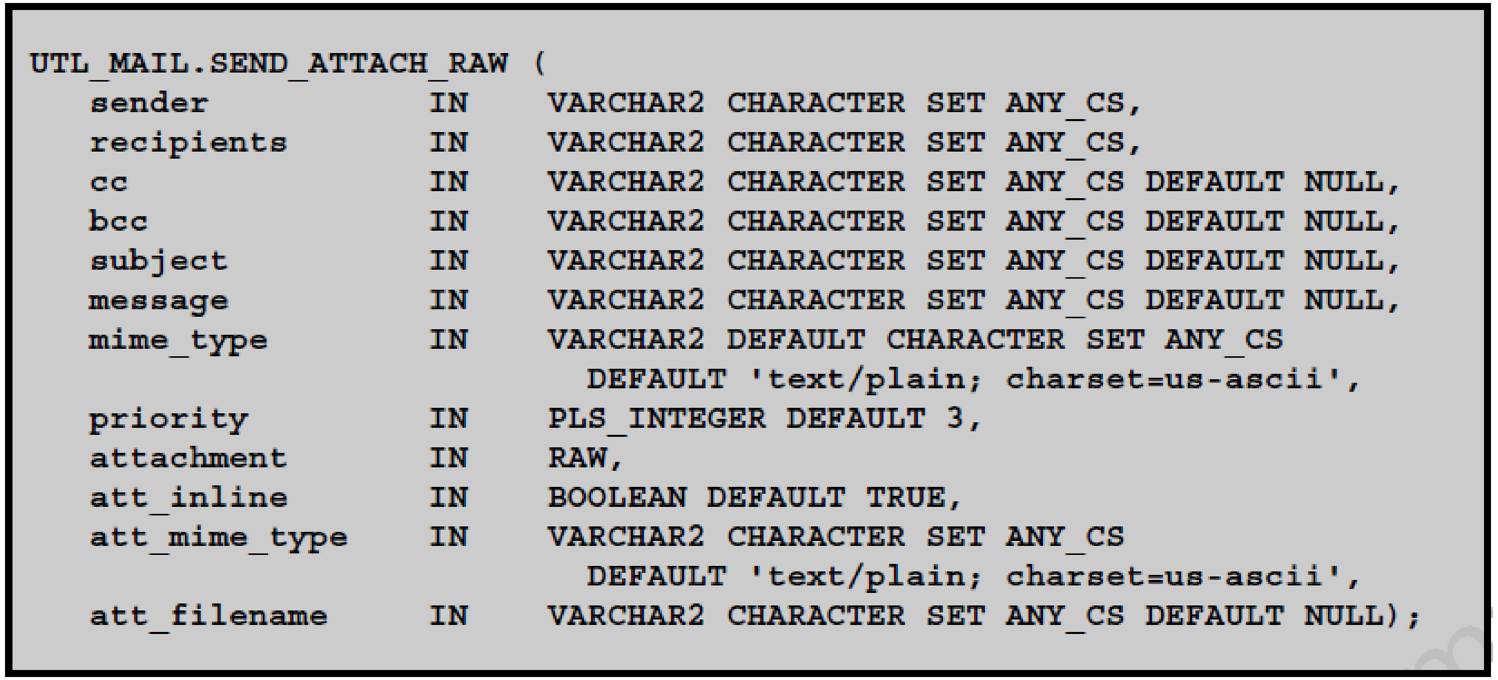


```
UTL_MAIL.SEND (
   sender       IN    VARCHAR2 CHARACTER SET ANY_CS,
   recipients   IN    VARCHAR2 CHARACTER SET ANY_CS,
   cc           IN    VARCHAR2 CHARACTER SET ANY_CS
                        DEFAULT NULL,
   bcc          IN    VARCHAR2 CHARACTER SET ANY_CS
                        DEFAULT NULL,
   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);
```

MIME:Multipurpose Internet Mail Extensions

The SEND_ATTACH_RAW Procedure

This procedure is the SEND procedure overloaded for RAW attachments.

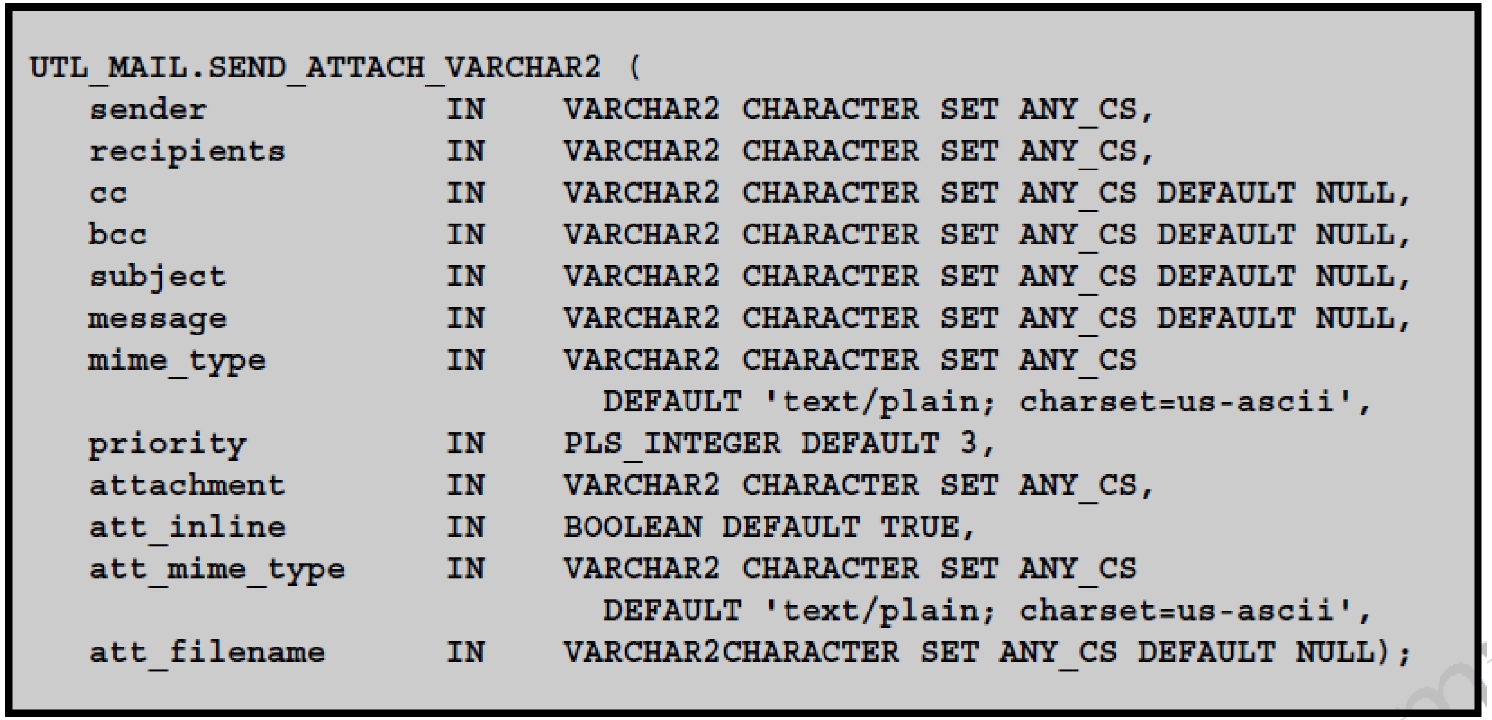


```
UTL_MAIL.SEND_ATTACH_RAW (
   sender            IN    VARCHAR2 CHARACTER SET ANY_CS,
   recipients        IN    VARCHAR2 CHARACTER SET ANY_CS,
   cc                IN    VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
   bcc               IN    VARCHAR2 CHARACTER SET ANY_CS DEFAULT NULL,
   subject           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);
```

MIME:Multipurpose Internet Mail Extensions

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);
```

MIME:Multipurpose Internet Mail Extensions

- Thank You