



PL/PDF PDF file distribution via e-mail

v2.3.1

pdf_mail procedure uses maildemo.sql package.

```
create or replace procedure pdf_mail(  
  p_sender  varchar2, — sender, example: 'Me <me@apple.com>'  
  p_recipients varchar2, — recipients, example: 'Someone <someone@pear.com>'  
  p_subject  varchar2, — subject  
  p_text     varchar2, — text  
  p_filename varchar2, — name of pdf file  
  p_blob     blob     — pdf file  
  ) is
```

```
  conn  utl_smtp.connection;  
  i number;  
  len number;
```

BEGIN

```
  conn := demo_mail.begin_mail(  
    sender    => p_sender,  
    recipients => p_recipients,  
    subject   => p_subject,  
    mime_type => demo_mail.MULTIPART_MIME_TYPE);  
  demo_mail.begin_attachment(  
    conn      => conn,  
    mime_type => 'application/pdf',  
    inline    => TRUE,  
    filename  => p_filename,  
    transfer_enc => 'base64');
```

```
  — split the Base64 encoded attachment into multiple lines  
  i := 1;  
  len := DBMS_LOB.getLength(p_blob);
```

```
  WHILE (i < len) LOOP  
    IF(i + demo_mail.MAX_BASE64_LINE_WIDTH < len)THEN  
      UTL_SMTP.Write_Raw_Data (conn  
        , UTL_ENCODE.Base64_Encode(  
          DBMS_LOB.Substr(p_blob, demo_mail.MAX_BASE64_LINE_WIDTH, i)));  
    ELSE  
      UTL_SMTP.Write_Raw_Data (conn  
        , UTL_ENCODE.Base64_Encode(  
          DBMS_LOB.Substr(p_blob, (len - i)+1, i)));  
    END IF;
```

```
  UTL_SMTP.Write_Data(conn, UTL_TCP.CRLF);  
  i := i + demo_mail.MAX_BASE64_LINE_WIDTH;  
  END LOOP;
```



```
demo_mail.end_attachment(conn => conn);
```

```
demo_mail.attach_text(  
conn    => conn,  
data    => p_text,  
mime_type => 'text/html');
```

```
demo_mail.end_mail( conn => conn );
```

```
END;  
/
```