PL/SQL: Built-in I/O package

Built-in Packages

Some of the PL/SQL functionality is provided in a form built-in libraries or packages of functions and procedures. A brief list of PL/SQL's built-in packages is below:

**dbms_output package**: provides functionality for I/O operations.

**dbms_sql package**: provides ability to execute dynamically created SQL statements (including relational schema alterations).

**http package**: HTTP and HTML-related procedures.

### dbms_output package

**dbms_output package** includes procedures and functions necessary for I/O operations.

**i/o buffer**: **dbms_output package** works with an i/o buffer. The **put** procedures add content to the buffer, the **get** procedures retrieve content from the buffer.

**Enabling buffer output.** This can be done in two ways.

- **SQL*plus.** In sql*plus type

  ```sql
  SQL> set serveroutput on
  ```

  to enable the output with the default buffer size of 2000. To change buffer size, use

  ```sql
  SQL> set serveroutput on size <value>
  ```

- **PL/SQL.** Use one of the following procedure calls:

  - `dbms_output.enable` : enables output with default buffer size
  - `dbms_output.enable(<value>)` : enables output with specified buffer size
Disabling buffer output. Can be done in two ways as well:

- **SQL*plus:**

  SQL> set serveroutput off

- **PL/SQL:**

  dbms_output.disable;

Putting information into buffer. The following procedures put information into the buffer:

<table>
<thead>
<tr>
<th>procedure</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>new_line</td>
<td>inserts line-feed/carriage return sequence into the buffer</td>
</tr>
<tr>
<td>put(&lt;value&gt;)</td>
<td>inserts given value into the buffer</td>
</tr>
<tr>
<td>put_line(&lt;value&gt;)</td>
<td>inserts &lt;value&gt; into the buffer, follows with lf/cr</td>
</tr>
</tbody>
</table>

Retrieving information from the buffer. Information can be extracted from the buffer. The buffer is treated as a **queue**, that is, information is extracted in a **FIFO** (first in – first out) order.

<table>
<thead>
<tr>
<th>procedure</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>get_line(line out varchar2, status out integer)</td>
<td>retrieves one line from the buffer</td>
</tr>
<tr>
<td></td>
<td>retrieved data stored in line;</td>
</tr>
<tr>
<td></td>
<td>status=0 if success</td>
</tr>
<tr>
<td>get_lines(lines out string255_table, nlines in out integer)</td>
<td>retrieves specified number of lines into a string table lines.</td>
</tr>
<tr>
<td></td>
<td>nlines - number of lines retrieved</td>
</tr>
</tbody>
</table>

`string255_table` is defined in the package as:

```sql
type string255_table is table of varchar(255) index by binary_integer;
```