

INTERNET ENABLED SQL*PLUS

Luan Nim, Oracle Corporation

INTRODUCTION

In today's rapidly changing world, new technologies appear faster than anyone can imagine. More and more applications are developed with a graphical user interface in mind. There are few Oracle tools that are still command line based — SQL*Plus is the most commonly used.

SQL*Plus is the command line interface to the Oracle database, and it has been around a long time. SQL*Plus is a Client/Server application that allows you to enter and execute SQL statements and PL/SQL blocks quickly and easily, it also has reporting capabilities, and is able to perform database administration tasks.

With Internet enabled applications fast gaining popularity with software developers, SQL*Plus must provide remote access to Oracle databases using tools such as web browsers. SQL*Plus needs to change to be part of this exciting Internet future.

This paper discusses the following:

- How SQL*Plus provides options to produce reports in HTML format and view the reports using a web browser.
- How SQL*Plus can be run from CGI scripts on a web server. This allows you to produce reports on demand and display the output in a web browser.
- Where SQL*Plus is heading in future releases.

HISTORY OF SQL*PLUS

SQL*Plus has been around for a long time, pretty much since the beginning of Oracle. The original name of the product was not SQL*Plus, but UFI (User Friendly Interface).

As features such as procedural capabilities, IF statements and looping capabilities were planned to be added to UFI, the name was subsequently changed to AFI (Advanced Friendly Interface). This name reflected the advancement of the product. The planned procedural enhancements never made to the production release as Oracle released PL/SQL, which included the planned AFI functionality.

The name AFI lives on in the name of the temporary file created when you use the SQL*Plus EDIT command. That file is named AFIEDT.BUF.

With the emergence of the Internet, telecommuting is gaining wide acceptance as the future way to work. SQL*Plus must provide ways to perform tasks remotely to enable this and many other new ways of working.

CLIENT/SERVER SQL*PLUS

Today, SQL*Plus is widely used throughout the world by software developers and novice users to access Oracle databases. The most common task is to use ad hoc queries to access information quickly and easily from Oracle databases. Another task for which SQL*Plus is often used is to run scripts during the installation of Oracle products.

As Server Manager is becoming obsolete (from Oracle8i release 8.2 onwards), SQL*Plus will be the *only* command line tool able to interact with an Oracle database.

SQL*Plus will always have a place in the Client/Server environment, even with the introduction of Internet enabled SQL*Plus. The emphasis on SQL*Plus as a Client/Server tool is still as important to Oracle as it always has been. Internet enabled SQL*Plus is another feature within SQL*Plus, but not the only feature. Many users will still want to use SQL*Plus as a 'fat' client.

INTERNET ENABLED SQL*PLUS

The SQL*Plus development team have just completed the first stage of a project which will Internet enable SQL*Plus. The rollout of the project will take place over releases 8.1.6, 8.1.7 and future releases of SQL*Plus.

ARCHITECTURE

Internet enabled SQL*Plus consists of three architectural components (three tier).

- SQL*Plus user interface
- SQL*Plus engine
- Oracle8i

The following diagram shows the breakdown of the three tiers.

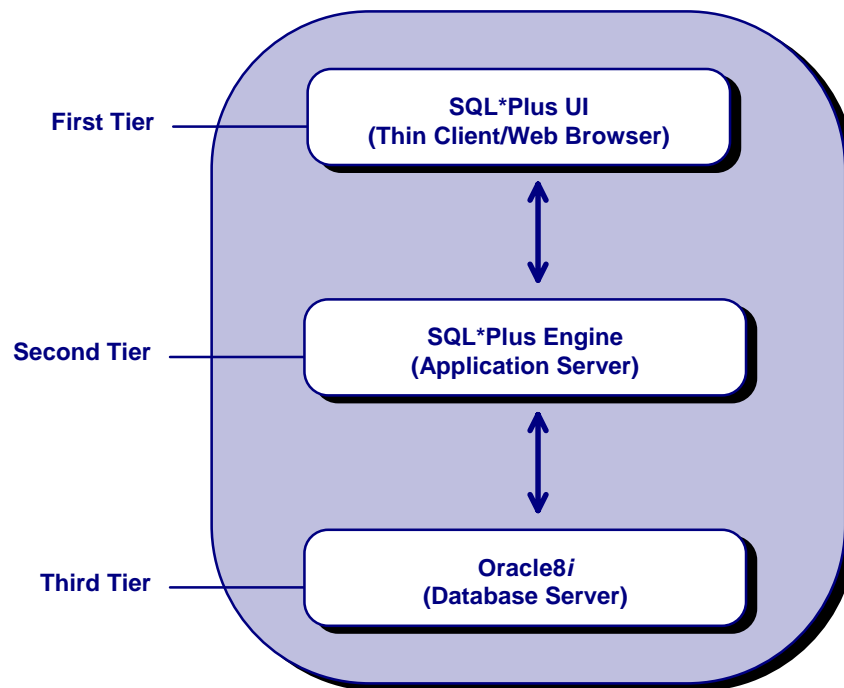


Figure 1. Architecture of Internet Enabled SQL*Plus

SQL*PLUS USER INTERFACE

The first tier is the SQL*Plus user interface, or thin client. The first tier is essentially a web browser running on any machine. You can use any web browser to access SQL*Plus as long as you have access to the Internet or an intranet.

SQL*PLUS ENGINE

The second tier is the SQL*Plus engine. This tier communicates between SQL*Plus and the Oracle database server. SQL*Plus processes requests from the thin client (first tier), accesses the database server (third tier), and returns the appropriate data to the thin client.

The communication between the SQL*Plus engine and the database server can be viewed as:

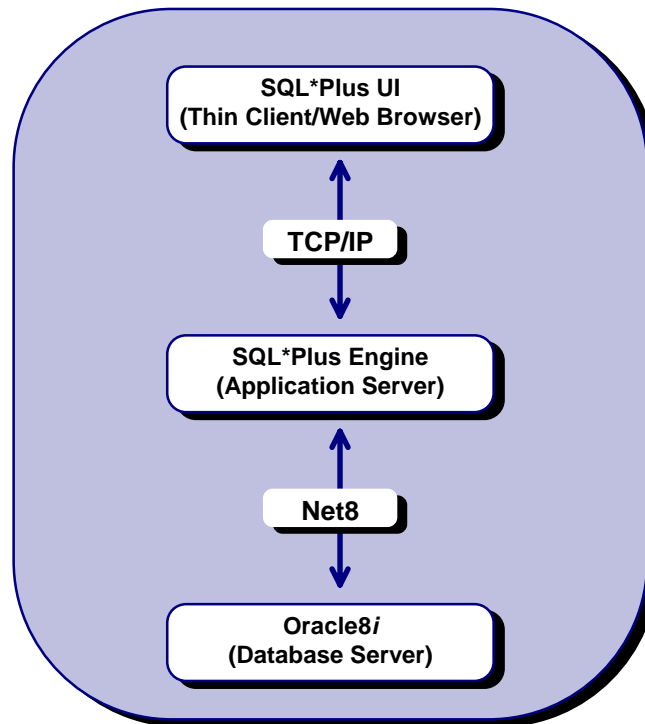


Figure 2. Internet Enabled SQL*Plus Communications

ORACLE8i

The third tier is, of course, Oracle8i. Oracle8i is the latest version of Oracle's industry leading database, the database for Internet Computing. Oracle8i is the most complete and comprehensive platform for building, deploying and managing Internet applications. Oracle8i provides the lowest cost platform for developing and deploying applications on the Internet.

SQL*PLUS 8.1.6

SQL*Plus release 8.1.6 can produce reports in HTML format. The report data is wrapped in the HTML <pre> (preformatted) tags. In future releases of SQL*Plus, features may be added to extend this functionality.

SQL*Plus release 8.1.6 also allows you to run SQL*Plus from CGI scripts on a web server. This allows you to dynamically produce reports and display the output in a web browser.

To view HTML reports produced by SQL*Plus, you must have a web browser compatible with HTML version 3.2 (or higher).

DESIGN REQUIREMENTS

Before the SQL*Plus development team started the design of the first phase of the Internet enabled SQL*Plus project, the design requirements needed to be discussed. The design requirements turned out to be:

- SQL*Plus must support the multitude of scripts already in existence without reverse engineering scripts.
- Whatever web markup language was decided upon, it had to be simple to use, and supported by all popular web browsers.
- New commands must be able to be expanded so that new options can be added in future releases.
- New commands must allow for user defined HEAD and BODY attributes, so that cascading style sheet information, or any other text, could be included.

- Key words in new commands must be similar to the reserved words used in HTML, so that they could be easily understood and remembered.
- When an HTML report is generated, the report must be in valid HTML format.
- This feature should produce embedded reports, that is, no HTML tags, only the general tags around data. This is for users who only want current data from the database as they have special requirements in their HTML tags.
- This feature must be enabled in standard, file based and a server side CGI environments.

INTERACTIVE SESSION

You can use Internet enabled SQL*Plus in an interactive session to create HTML output. The output is spooled to a file, and can be viewed with a web browser.

Start SQL*Plus as usual, and create SQL*Plus script, for example:

```
@employee.sql
```

Note that you do not have to create a SQL*Plus script. You can also create HTML output by simply typing commands at the SQL*Plus prompt, without saving the commands to a file.

Enter the commands to create HTML output. Your file should look something like this:

```
SET MARKUP HTML ON SPOOL ON HEAD '<title>Employee List</title> -
<style type="text/css"> <!-- BODY {background: red} --> </style>'
SET ECHO OFF
SPOOL employee.html
SELECT empno, ename from emp;
EXIT
```

Note that the minus sign (“-”) at the end of the first line is the SQL*Plus continuation character.

The results of this query are spooled to an HTML file, which you can view with your web browser.

SHELL SCRIPT

You can write a Shell script to create an HTML file with the report output embedded within the HTML file. The output is spooled to a file and can be viewed with a web browser.

Create an operating system file, for example *employee.sh*. Your script should look something like this:

```
ECHO "<HTML><HEAD><TITLE>Employee List</TITLE></HEAD><BODY><PRE> " >
  $ORACLE_HOME/sqlplus/employee.html
sqlplus scott/tiger @$ORACLE_HOME/sqlplus/employee.sql >>
  $ORACLE_HOME/sqlplus/employee.html
ECHO "</PRE></BODY></HTML>" >> $ORACLE_HOME/sqlplus/employee.html
```

Create a SQL*Plus script, for example, *employee.sql*. Your script should look something like this:

```
SET MARKUP HTML ON
SET ECHO OFF
SELECT ENAME, EMPNO FROM EMP;
EXIT
```

The results of the Shell script are written to *employee.html*, which can be viewed with your web browser.

CGI SCRIPT

You can use Internet enabled SQL*Plus as a CGI script to create HTML output from a web browser.

1. Create a CGI script. (*sqlscript.sh*).
2. Create a SQL script. (*employee.sql*).
3. Save these to the *cgi-bin* directory of your web server.
4. Run the CGI script.

CGI SCRIPT (*SQLSCRIPT.SH*)

Create a CGI script, for example, *sqlscript.sh*. Your script should look something like this:

```
#!/bin/sh

echo Content-type: text/html
# Required blank line
echo
# Your ORACLE_HOME
ORACLE_HOME=/vobs/oracle
# Your ORACLE_SID
ORACLE_SID=plus81
# Your Net8 TNSNAMES.ORA path if required for remote databases
# TNS_ADMIN=/var/opt/oracle
export TNS_ADMIN
export ORACLE_HOME
export ORACLE_SID

# SQL*Plus executable path
# and enable SQL*Plus markup feature and include Net8 service name
# and SQL script path and script name
$ORACLE_HOME/bin/sqlplus -M "html on" scott/tiger@connect_string
@$ORACLE_HOME/sqlplus/employee.sql
```

SQL*PLUS SCRIPT (*EMPLOYEE.SQL*)

Create a SQL*Plus script, for example, *employee.sql*. Your script should look something like this:

```
SELECT empno, ename from emp;
EXIT
```

To run the CGI script and view the results of the query in your browser, enter the URL which points to your script in your web browser, for example

```
http://webserver.domain.com/cgi-bin/sqlscript.sh
```

SET MARKUP COMMAND

To use Internet enabled SQL*Plus, you must use the new SET command, SET MARKUP.

SYNTAX

```
SET MARKUP HTML [ON|OFF] [HEAD text] [BODY text]
                [ENTMAP {ON|OFF}] [SPOOL {ON|OFF}]
                [PRE[FORMAT] {ON|OFF}]
```

One of the most important clauses of the SET MARKUP command is the SPOOL clause.

```
SPOOL {ON|OFF}
```

Controls the writing of tags to the spool file. SPOOL ON enables <HTML> and </HTML> tags to be written to the spool file. The SPOOL ON option does not take effect until the SQLPLUS SPOOL command is entered. SPOOL OFF disables the SPOOL option.

EXAMPLE 1, CREATING AN HTML FILE IN AN INTERACTIVE SESSION

This example walks you through the creation of the HTML output in an interactive SQL*Plus session. The HTML output will be spooled to a file, and then you can view it with a web browser.

1. Start and log in to SQL*Plus. At the SQL prompt, enter the SET MARKUP command, with the appropriate options.

```
SET MARKUP HTML ON SPOOL ON HEAD '<title>Example</title> <style -
type="text/css"> <!-- BODY {background: red} --> </style>'
```

You can use the HEAD option to include an embedded style sheet, or any other viable HEAD text. Note that the minus sign (“-”) at the end of the first line is the SQL*Plus continuation character.

2. Enter the SPOOL command with the spool file name, in this case, *example.html*:
SPOOL example.html

After the SPOOL command, anything entered or displayed on the standard output is written to spool file.

SET MARKUP HTML ON SPOOL ON tells SQL*Plus to enable the HTML feature and write HTML tags to a spool file. The SPOOL command triggers the <HTML>, <HEAD>, <BODY> and <PRE> tags to be written to the spool file, followed by the data from the query.

3. Enter a query, for example
SELECT empno, ename from emp;
4. Enter the SPOOL OFF command, or the EXIT command.
SPOOL OFF

The </PRE>, </BODY> and </HTML> tags are written to the spool file when you enter either the SPOOL OFF or EXIT command.

5. Start your web browser and open the *example.html* file.

Your screen should look like this:

```
SET MARKUP HTML ON SPOOL ON HEAD '<title>Example</title> <style
type="text/css">
<!-- BODY {background: red} --> </style>'
SPOOL example.html
SELECT empno, ename from emp;
SPOOL OFF
```

Your HTML output file *example.html* should look like this:

```
<html>
<head>
<title>
Example
</title> <style type="text/css"> <!-- BODY {background: red} -->
</style>
<meta Name="generator" content="SQL*Plus 8.1.6">
</head>
<body >
<pre>
SQL> SELECT empno, ename from emp;
```

ENAME	EMPNO
SMITH	7369
ALLEN	7499
WARD	7521
JONES	7566
MARTIN	7654
BLAKE	7698
CLARK	7782
SCOTT	7788
KING	7839
TURNER	7844
ADAMS	7876
JAMES	7900
FORD	7902
MILLER	7934

14 rows selected.

```
SQL> SPOOL OFF
</pre></body></html>
```

EXAMPLE 2, CREATING A REPORT EMBEDDED IN AN HTML FILE FROM A SHELL SCRIPT

This example walks you through creating a report, which is embedded in an HTML file, from a Shell script. The embedded report generated from SQL*Plus is appended to the HTML file and a subsequent shell command is used to write the </HTML> tag to the end of the HTML file.

1. Create a Shell script using your usual text editor. Your script can be written in Perl, Java or any other language compatible with your web server. A Shell script is used in this example.
2. Add the following text to your Shell script and edit it to suit your configuration:

```
#!/bin/sh
# The <HTML> tag is generated using the ECHO command and redirected
# to a an HTML file.
echo "<HTML><HEAD><TITLE>Employee List</TITLE></HEAD><BODY><PRE>" >
$ORACLE_HOME/sqlplus/employee.html
# The data is retrieved from the database using SQL*Plus and
# redirected to an HTML file
sqlplus -s scott/tiger @$ORACLE_HOME/sqlplus/employee.sql >>
$ORACLE_HOME/sqlplus/employee.html
# The </HTML> tag is generated using the ECHO command and redirected
# to an HTML file.
Echo "</PRE></BODY></HTML>" >> $ORACLE_HOME/sqlplus/employee.html
```

3. Save this script as *employee.sh* in *\$ORACLE_HOME/sqlplus*

4. Create a SQL*Plus script in your text editor which contains:

```
SET MARKUP HTML ON
SET ECHO OFF
SELECT EMPNO, ENAME FROM EMP;
EXIT
```

Save the script as *employee.sql* in *\$ORACLE_HOME/sqlplus*

5. Run the Shell script *employee.sh* at the operating system level, for example.

```
$employee.sh
```

The Shell script *employee.sh* creates the *employee.html* file that contains the <HTML> tags, then starts SQL*Plus and runs the *employee.sql* script. The results are stored in *employee.html*. You can view the result from a web browser.

Your output should look like this:

```
<HTML><HEAD><TITLE>Department Report</TITLE></HEAD><BODY><PRE>
```

```
SQL*Plus: Release 8.1.6.0.0 - Production on Mon Oct 4 15:06:53 1999
```

```
(c) Copyright 1999 Oracle Corporation. All rights reserved.
```

```
Connected to:
```

```
Oracle8i Enterprise Edition Release 8.1.6.0.0 - Production
```

```
With the Partitioning and Java options
```

```
PL/SQL Release 8.1.6.0.0 - Production
```

```
-----
EMPNO ENAME
-----
 7369 SMITH
 7499 ALLEN
 7521 WARD
 7566 JONES
 7654 MARTIN
 7698 BLAKE
 7782 CLARK
 7788 SCOTT
 7839 KING
 7844 TURNER
 7876 ADAMS
```

```

EMPNO ENAME
-----
7900 JAMES
7902 FORD
7934 MILLER

```

14 rows selected.

```

Disconnected from Oracle8i Enterprise Edition Release 8.1.6.0.0 -
Production
With the Partitioning and Java options
PL/SQL Release 8.1.6.0.0 - Production
</pre></body></html>

```

EXAMPLE 3, CREATING AN HTML FILE FROM A CGI SCRIPT

This example walks you through creating a CGI script to run SQL*Plus as a CGI program, producing a report from an existing SQL script, and viewing the results in a web browser.

1. Create a Shell script using your usual text editor. Your script can be written in Perl, Java or any other language compatible with your web server. A Shell script is used in this example.
2. Add the following text to your script and edit it to suit your configuration:

```

#!/bin/sh

echo Content-type: text/html
# Required blank line
echo
# Your ORACLE_HOME
ORACLE_HOME=/vobs/oracle
# Your ORACLE_SID
ORACLE_SID=plus81
# Your Net8 TNSNAMES.ORA path if required for remote databases
# TNS_ADMIN=/var/opt/oracle
export TNS_ADMIN
export ORACLE_HOME
export ORACLE_SID

# SQL*Plus executable path
# and enable SQL*Plus markup feature and include Net8 service name
# and SQL script path and script name
$ORACLE_HOME/bin/sqlplus -M "html on" scott/tiger@connect_string
@$ORACLE_HOME/sqlplus/employee.sql

```

3. Save this script as *sqlscript.sh* in the *cgi-bin* directory of your web server.
4. Create a SQL*Plus script in your text editor which contains:

```

SELECT ename from emp;
EXIT;

```

Save the script as *employee.sql* in *\$ORACLE_HOME/sqlplus*

5. Start your web browser and enter the URL appropriate for your web server: It should be similar to:

```
http://webserver.domain.com/cgi-bin/sqlscript.sh
```

The shell script *sqlscript.sh* runs, which in turn starts SQL*Plus and runs the *employee.sql* script. The results are displayed in your web browser.

Your output should look like this:

```
SQL*Plus: Release 8.1.6.0.0 - Production on Wed Sep 29 16:58:18 1999
```

(c) Copyright 1999 Oracle Corporation. All rights reserved.

```

Connected to:
Oracle8I Server Release 8.1.6.0.0 - Production

```


With the distributed, parallel query and Spatial Data options
 PL/SQL Release 8.1.6.0.0 - Production

```

ENAME
-----
SMITH
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS

```

```

ENAME
-----
JAMES
FORD
MILLER

```

14 rows selected.

Disconnected from Oracle8i Server Release 8.1.6.0.0 - Production
 With the distributed, parallel query and Spatial Data options
 PL/SQL Release 8.1.6.0.0 - Production

SQL*PLUS 8.1.7

New features in SQL*Plus release 8.1.7 will focus heavily on HTML data presentation. We have planned to add the following feature to SQL*Plus release 8.1.7:

- Change the default HTML tag for SQL*Plus web output from the <pre> tag to the <table> tag. This will enable SQL*Plus web output to be displayed in an HTML table.

SQL*PLUS FUTURE RELEASES

The goal for future releases of SQL*Plus is to have full three tier functionality. That is, to

- Use any popular web browser to access SQL*Plus
- Submit SQL commands from a web browser
- Retrieve data from an Oracle database through a web browser
- Display output to a web browser

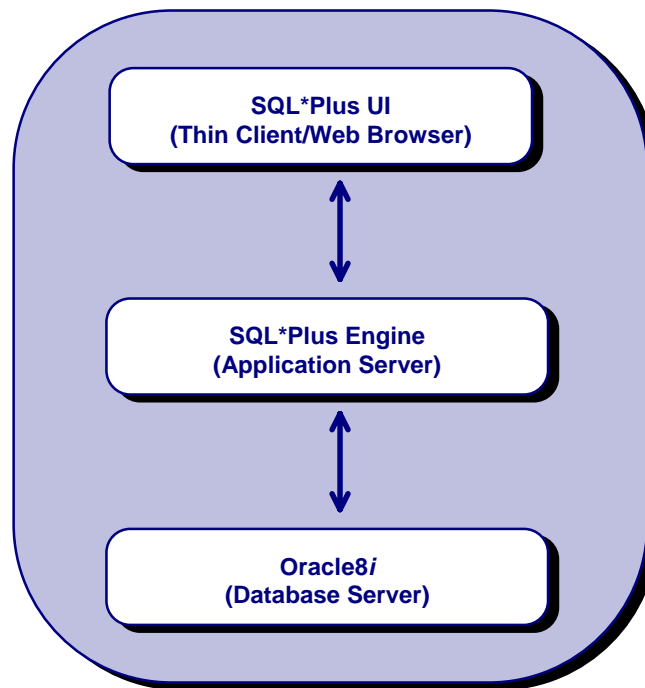


Figure 3. Internet Enabled SQL*Plus

BENEFITS OF INTERNET ENABLED SQL*PLUS

- Reduce total cost of ownership
- Centralize SQL scripts
- Remove requirement to install SQL*Plus and Net8 on PCs
- Access SQL*Plus from anywhere with a web browser

THE INTERNET CHANGES EVERYTHING

The Internet has definitely changed SQL*Plus from a Client/Server tool to an Internet enabled tool. As the new millennium unfolds, it is hoped that the Internet enabled SQL*Plus tool will become as widely used as SQL*Plus on Client/Server.