PL/SQL Developer 7.0
New Features

December 2005
Contents

CONTENTS ..............................................................................................................................................................3

1. INTRODUCTION......................................................................................................................................5

2. DIAGRAM WINDOW....................................................................................................................................6
   2.1 CREATING A DIAGRAM .....................................................................................................................6
   2.2 SAVING AND OPENING A DIAGRAM FILE .........................................................................................8
   2.3 UPDATING A DIAGRAM .......................................................................................................................8

3. GRAPH WINDOW ...........................................................................................................................................9

4. ODBC IMPORTER .......................................................................................................................................11

5. COMPARE TABLE DATA .............................................................................................................................13

6. TEST MANAGER .........................................................................................................................................15
   6.1 CREATING A TEST SET ......................................................................................................................16
   6.2 RUNNING A TEST SET .........................................................................................................................17

7. DOCKABLE & FLOTTING TOOLS ..............................................................................................................18

8. SEARCH BAR ..............................................................................................................................................19

9. PROGRAM WINDOW ENHANCEMENTS ...............................................................................................20
   9.1 NAMING CONVENTIONS ....................................................................................................................20
   9.2 HIGHLIGHT VARIABLES .....................................................................................................................21
   9.3 SAVE WINDOW STATE PREFERENCE ...............................................................................................21
   9.4 SUBSTITUTION VARIABLES ..............................................................................................................21
   9.5 REFACTORING .....................................................................................................................................21
   9.6 CODE CONTENTS LAYOUT ..................................................................................................................22

10. SQL WINDOW ENHANCEMENTS ...........................................................................................................23
    10.1 MULTIPLE STATEMENTS ....................................................................................................................23
    10.2 GRAPH WINDOW ...............................................................................................................................23
    10.3 ALTERNATE GRID ROW COLORS ..................................................................................................24
    10.4 GRID COLUMN TOTALS ....................................................................................................................24
    10.5 GRID SELECTION INDICATOR ..........................................................................................................25
    10.6 EXCEL EXPORT ................................................................................................................................25

11. COMMAND WINDOW ENHANCEMENTS ...............................................................................................26

12. SESSION WINDOW ENHANCEMENTS ....................................................................................................27
    12.1 SESSION FILTERS ...............................................................................................................................27
    12.2 CONCATENATE HINT ..........................................................................................................................28
    12.3 CLOB AND LONG COLUMNS .............................................................................................................28

13. EDITOR ENHANCEMENTS ..........................................................................................................................29
    13.1 GLOBAL NAVIGATION ........................................................................................................................29
    13.2 GLOBAL BOOKMARK LIST ................................................................................................................29
    13.3 COLOR MARKS ....................................................................................................................................29
    13.4 SELECTION SUBMENU .......................................................................................................................30
    13.5 PREFERENCES ....................................................................................................................................30

14. GRID ENHANCEMENTS .............................................................................................................................31
    14.1 OBJECT GRID FILTERS .....................................................................................................................31
    14.2 FIND FUNCTION .................................................................................................................................31
15. FILES > FORMAT PREFERENCE PAGE ................................................................. 32
16. ACTIVE QUERY BUILDER PLUG-IN ................................................................. 33
17. ORACLE FILE SYSTEM (OFS) ........................................................................... 34
   17.1 OFS MANAGER .......................................................................................... 34
   17.2 OFS USAGE .............................................................................................. 36
18. PLUG-IN INTERFACE ....................................................................................... 38
19. MISCELLANEOUS ENHANCEMENTS .............................................................. 39
1. Introduction

This document describes the most important new features in PL/SQL Developer 7.0. For a detailed description of previous features, see the PL/SQL Developer 6.0 User’s Guide.
2. Diagram Window

The Diagram Window allows you to create a graphical representation of a selection of objects. This way you can easily visualize (a part of) the database objects of your application or project and their relations:

You can add all relevant object types to a diagram, and include the relations between them. Foreign key relations between tables are automatically included. For large diagrams you can add additional structure by including object groups.

A diagram can be used for documentation purposes, but can also serve as a workspace. Right-clicking on an object provides access to all object functions, and double-clicking on an object will invoke the default action for the object type.

2.1 Creating a diagram

To create a new diagram, select the Diagram Window from the New submenu in the File menu. An empty Diagram Window will appear. To add a database object to the diagram, drag it from the Object Browser into the Diagram Window. You can drag & drop multiple objects at once.

To move a diagram object, click on the title bar and drag it to the new location. To resize a diagram object you can drag an edge or a corner. To delete an object from the diagram, press on the X at the upper-right of the object. This will only affect the diagram, and the database object will not be dropped.

Showing and hiding items

To show or hide a specific item of a diagram object, right-click on it and go to the Items submenu. Here you will see a submenu with all items that are applicable for the selected object. For a table you can select which items of a column, key or index you want to show. You can also hide the complete key or index section.

By default all items will be displayed, but if you right-click on an object and select Set as default, the visible items will apply to all objects added to a diagram in the future. You can do this for all object types.
Adding relations

To add a relation between 2 objects, click on the icon of an element of the source object, and drag it to the element of the target object. A line will be drawn between the 2 elements of the objects. You can move the line to another element by selecting the line and moving the selection point up or down. To remove a line, select it and press the Delete key.

If you add a table to a diagram, foreign key relations with other tables on the diagram will automatically be included. These relations can be deleted if necessary.

By default relations will be drawn as straight lines. You can add additional bends by right-clicking on a line and selecting Insert bend from the popup menu. You can also hold down the Ctrl key while clicking on a line to insert a bend. A new selection point is added to the line as a result.

To create a one-to-many relation, right-click on the end of the line and select Change line end from the popup menu.

Adding groups

To add additional structure to a large diagram, you can create object groups. Groups can have distinctive colors, and can have a title at the upper or lower left or right:

To add a group, right click at an empty position on the diagram and select Add group from the popup menu. A new group will appear at this location, which you can resize to appropriate dimensions. All objects that are completely covered by the group will implicitly be part of that group, and moving the group will also move these enclosed objects. If you hold down the Ctrl key while moving a group, the enclosed objects will not be moved.
To change the appearance of a group, right-click on it and select the Properties item from the popup menu. You can also double-click on the group. The following screen will appear:

You can change the font, fill color, line color and line size of the group, and make the group transparent. You can optionally enter the title of the group, and specify the location of the title.

To create a default group appearance, right-click on it and select Set as default. This will apply to all groups added to a diagram in the future.

To delete a group from the diagram, select it and press the Delete key. This will only delete the group, and not the enclosed objects.

2.2 Saving and opening a diagram file

To save a diagram to a file, select the Save item from the File menu. A diagram will be saved in a file with a .pdd extension, which can be reopened later. To define a standard location for diagram files, go to the Directories preference page and set the corresponding option.

2.3 Updating a diagram

A diagram is a static representation of the database. If database objects change over time, you will need to update the diagram to reflect these changes. To do so, right-click on a diagram object and select Update from DB from the popup menu. To update all objects on the diagram, right-click on it and select Update All from DB.

If an object is dropped from the database, you will also need to delete it from the diagram. If an object is renamed in the database, you will need to delete the diagram object and add it with the new name.
3. **Graph Window**

The Graph Window can be used to quickly visualize column data of a SQL Window or Report Window. Consider the following query:

```sql
select deptno, count(*) as employees, sum(sal) as salary
from emp
group by deptno
```

To see a graph of the salaries of the department you can press the *Graph* button:

This will start the *Graph Wizard*, which will first allow you to select the data for the X-axis:

This shows all columns and an *[Index]* item, which is useful if you don’t have an X-axis value but want to display data on the Y-axis in order of occurrence. In this case we want to display the salary by department number, so we select *DEPTNO* and press the *Next* button to select the data for the Y-axis:

After selecting *SALARY* for the Y-axis, you can press *Next* to select the graph type:
After pressing Finish, the resulting Graph Window will be displayed:

The toolbar of the Graph Window provides the following functions:

- **Rotate** – Press the left mouse button and move the mouse to rotate the graph around the X and Y axis.

- **Move** – Press the left mouse button and move the mouse to move the graph across the window. This function is always available by pressing the right mouse button, regardless of the activated toolbar function.

- **Zoom** – Press the left mouse button and move the mouse to zoom in to and out of the graph.

- **Depth** – Press the left mouse button and move the mouse to increase or decrease the depth of the graph.

- **3D** – Switch the graph between a 2D and 3D view.

- **Edit** – Edit the graph data and layout.

- **Save as template** – Saves the layout of the graph as a template for future graphs based on the same column names. If you create a graph later and select From template from the Graph button popup menu, then the layout information from this template will be used.

You can additionally print a graph or copy it to the clipboard by pressing the corresponding buttons on the main toolbar.
4. **ODBC Importer**

The ODBC Importer tool allows you to import data from any ODBC data source into an Oracle table:

![ODBC Importer Tool](image)

On the first page of the ODBC Importer you need to select the ODBC data source, and optionally provide a user name and password for this data source. You can subsequently connect, after which the table list will be populated with all tables from the data source. Selecting a table will populate the Result Preview pane. Instead of selecting a table, you can also provide a query by selecting the Import Query Result option and entering a query text. This can be useful if you want to limit the rows or columns, or if the results come from more than one table.
After entering the source specifications, you can switch to the *Data to Oracle* tab page to select the destination table and to specify the column mapping:

In the *General* section you can select an owner and a table into which the records should be imported. You can also select if you want to clear the table before the import, select a commit interval (0 = commit at end) and to ignore duplicate rows or to overwrite them.

In the *Fields* section you can map fields from the ODBC data source (left) to columns in the Oracle table (right). For each column you can additionally supply a SQL function. For example, if you want to convert a string field to uppercase, enter `upper(#)` The hash sign will be replaced by the field data for each record, and the resulting expression will be supplied in the insert statement.

After completing the field mapping, you can press the *Import* button to import the data into the Oracle table. If you hold down the *Ctrl* key while pressing this button, an example insert statement will be displayed so that you can verify the field mapping and SQL functions you specified.

To reuse this import definition later, you can press the *Save Definition* button on the toolbar. This definition can be opened later by pressing the *Open Definition* button.
5. **Compare Table Data**

After inserting, updating and deleting records in one or more tables during development, it may be useful to propagate these changes to the same table owned by a different user. This may be a propagation from a development to a test database, or to a production database, or to another project member.

To do so, you can use the Compare Table Data tool from the Tools menu. This will bring up the following dialog:

On the Selection tab page you can select the table(s) you want to compare. After making this selection, you can press the Target Session button, to select the user and database that you want to compare. This will enable the Compare button, which you can press to start the compare operation, but you can also go to the Options tab page first:

On this page you specify various options for the compare process.

- **Actions** – You can select the various actions you want to be performed in the target session. If, for example, you omit the Delete action, only inserts and updates will be performed.
• **Commit every N records** – Controls after how many records a commit will occur. If you specify 0, only one commit will occur at the end of the process.

• **Mode** – Select Generate script to generate a SQL script with inserts, updates, deletes and commits. You can save this script later and run it in the Command Window or in SQL*Plus. Select Update database to immediately apply the changes in the target session.

• **Target schema** – If the tables are located in a different schema than the user of the target session, you can specify this schema here. The target user will need the necessary privileges on the selected tables in the target schema.

• **Include owner in SQL** – This option is only useful in script mode, and will prefix the table names in the generated SQL with the owner. Doing so will allow you to connect as a different user later when you run the script, and still apply the changes in the correct schema.

After selecting the appropriate options, you can press the Compare button to start the compare process. After completion, the Differences will be displayed:

In update mode, this tab page will merely display the total number of inserted, updated, and deleted records for each table.

In script mode, you can review all differences and save the resulting script by pressing the Save SQL to file button to the right. Note that you can limit the script to specific tables by making a table selection in the upper pane. You can also copy the SQL to the clipboard or apply the selected changes in the target session by pressing the corresponding buttons to the right.
6. **Test Manager**

To perform regression testing for your Oracle stored program units you can use the Test Manager. It allows you to define a Test Set, which is a collection of Test Scripts with input variable values and required output. It additionally allows you to specify required performance:

For this example the Test Set consists of 3 Test Scripts. For the selected `DeptName.tst` script the `p_empno` variable will get the value 7499 on input, and after execution the `result` variable value must be `SALES`.

Running a Test Set will quickly reveal if the tested program units still function correctly and/or with the required performance:

All failed Test Scripts will have a red indicator and will be placed at the top. In this case the `DeptName(7499)` script failed, because the specification required that the result is “SALES” instead of the actual “Sales” value.
6.1 Creating a Test Set

To create a Test Set, select the Test Manager item from the Tools menu. An empty Test Manager screen appears, for which you can create new Test Scripts or add existing Test Scripts. At the top of the Test Manager you see a list of Test Scripts, where you can add scripts to the set, remove scripts from the set, or change the order of the scripts. At the bottom you see the definition of the selected script.

Before creating a new Test Script, it is a good idea to save the Test Set by pressing the Save Test Set button on the toolbar. All new Test Scripts will be created in the same directory as the Test Set by default, and it may be a good idea to use a separate directory for each Test Set, so that all related test assets can be managed from this single directory.

Adding a Test Script to the Test Set

There are 2 methods to add a Test Script to the Test Set:

1. Drag & drop a procedure or function from the Object Browser to the Test Manager. A new Test Window will be opened with a call to this program unit, and variables for all parameters and the return value. Save the Test Script under a descriptive name, preferably in the same directory as the Test Set.

2. Press the New button to the right of the Test Script list. The following dialog will appear:

Click Yes to create a new, empty Test Script. You will need to manually program the Test Script and save it before it can be used in the Test Set.

Click No if you want to add an existing Test Script to the Test Set.

After the Test Script is saved, you can specify the definition in the Test Set:

- **Test script** – The filename of the Test Script. This will already by filled in if you have created and saved a new Test Script. If you want to add an existing Test Script, press the Browse button to the right of this field.

- **Description** – A functional description of the script. If you run a single script multiple times with different input variable values, you should include this in the description as well so that it can quickly be recognized in the list.

- **Enabled** – Use this option to enable or disable the script for the Test Set. Only enabled scripts will be included in the test run, and disabled scripts will be skipped.

- **Performance** – The required performance can be specified as the maximum number of seconds (fractions allowed). If you leave this field empty, the performance will not be tested.

- **Variables** – For each variable in the Test Script, enter the name, input value, and required output value. The input value will be passed to the Test Script before execution. The required output value will be tested after execution. If you leave the output value empty, it will not be tested. To test for null values, enter the word “null”.

To test for dbms_output, enter “dbms_output” for the variable name, and enter the required text for the output value.

At the lower left you see 2 buttons that allow you to quickly run and check the selected Test Script, or to open it in a Test Window for editing.
6.2 Running a Test Set

To run a Test Set, press the Run button on the toolbar. The Test Manager will switch to the Run tab page, and each enabled Test Script will be executed in a separate Test Window. Failures will be indicated with a red indicator, and will be placed at the top of the list. Successful scripts will have a green indicator, and will be placed at the bottom of the list.

To investigate a failure, right-click on the script and select Debug from the popup menu, or double-click on the script. A Test Window with the Test Script will be created, and the input variable values will be set accordingly. Now you can debug the Test Script to determine the cause of the error.

Right-click on the test run results to copy them to the clipboard, to print them, or to export them in various formats.
7. **Dockable & Floating Tools**

The following tools can be docked or can be presented in a floating window:

- Object Browser
- Window List
- Template List
- Bookmark List
- Search Bar
- Search Results

To dock a tool, simply drag it to the left, right, top or bottom of the work area. As soon as you come close to the side of the work area, a frame will be displayed, indicating the docking position. If another tool is already docked at this location, you can drag the new tool above, below, to the left, to the right, or within the other tool. If you drag one tool within the other, they will become available at the same position on different tab pages (see the Window List and Template List in the screenshot above).

To make a tool float, simply drag it from its docking position to a floating position.
8. Search Bar

The Search Bar is available from the Edit menu and can be displayed as a floating tool, or docked at the top or bottom of the work area. In its docked position, it can be permanently available to perform searches across multiple editors and multiple windows:

After entering a search word, you can press the Search button or press Enter. The search results will be highlighted in the editor(s), and you can press the Go to next/previous occurrence buttons to navigate the search results. You can press Esc to move the focus from the search bar to the editor.

The Search list button will bring up an hierarchical display of the search results:

In this tree view you can see the windows, editors, and lines where the search results are found. Clicking on a search result will bring the window and editor to the front, will navigate to the search result, and will select it. You can dock the search list to make it accessible at all times without blocking any other windows.

On the search bar you can additionally enable options to search all editors or just the current editor, to search case sensitive, to search for whole words only, and to use regular expressions.
9. Program Window Enhancements

The following enhancements have been made to the Program Window.

9.1 Naming conventions

The Hints preferences have been moved to a separate preference page, and a new Naming Conventions list has been added. In this list you can define naming conventions that will be checked when a program unit is compiled, or when the Show Compiler Hints function is invoked. A hint will be displayed for each element in the program unit source that does not meet these naming conventions.

The naming conventions list looks as follows:

<table>
<thead>
<tr>
<th>Element</th>
<th>Prefix</th>
<th>1st char</th>
<th>Allowed chars</th>
<th>Suffix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>p_</td>
<td>Any</td>
<td>Any</td>
<td></td>
<td>must start with 'p_'</td>
</tr>
<tr>
<td>Parameter</td>
<td>self</td>
<td>None</td>
<td>None</td>
<td></td>
<td>must be 'self'</td>
</tr>
<tr>
<td>PLSQL Type</td>
<td>T</td>
<td>Any</td>
<td>Any</td>
<td></td>
<td>must start with 'T'</td>
</tr>
</tbody>
</table>

For each naming convention you can define the element type (parameter, variable, and so on), the required prefix for the element type, the possible values for the first and subsequent characters (after the prefix), the required suffix, and a description for the hint that should be displayed when the naming convention is not met.

The description should be in the form as displayed in the example above. The actual hint message will be <Element> <Name> <Description>. For example: Parameter ‘StartDate’ must start with ‘p_’.

If an element must meet one of multiple naming conventions, then you can simply add multiple lines for the same element type. The descriptions of all lines will be displayed, separated with ‘or’. For the example above the parameter hint message will be:

Parameter 'StartDate' must start with 'p_' or must be 'Self'
9.2  Highlight variables

If the Program Window preference **Highlight variables** is enabled, then all occurrences of the variable under the cursor will be highlighted:

If the preference is disabled, you need to explicitly use the **Find Matches** function of the **Edit** menu.

9.3  Save window state preference

If the Program Window preference **Save window state** is enabled, then the following window state will be automatically be saved and restored for a file or database source:

- Window position & size
- Current editor
- Cursor position
- Code Contents size
- Bookmarks
- Color marks
- Substitution variable values

9.4  Substitution variables

If a program file contains one or more substitution variables (prefixed by an ampersand), then you can provide a value for these variables by pressing the & button at the lower left of the window. The substitution variables are replaced by their values in the PL/SQL source before it is sent to the server for compilation.

9.5  Refactoring

The refactoring function allows you to quickly reorganize your PL/SQL code. It works on the selected code, or – if no selection is made – on the current statement. Right-clicking on a statement or selection provides the following refactoring functions in the corresponding submenu:
• Rename item
Renames the current variable, parameter, constant or program unit. Both the declaration and usage will be renamed.

• Convert selection to procedure
If a program unit has become too large or too complex, you can make a selection and convert it to a separate procedure. All variables used within and outside of the selection will be converted to parameters. All variables used only within the selection will be moved from the current program unit to the new program unit. The selection will be replaced by a call to the new program unit.

• Convert selection to local constant
If a certain expression should be converted to a local constant, you can select it and provide the constant name. A local constant will be created within the current subprogram, of a type that is determined from the expression. All occurrences of the expression in the current subprogram will be replaced by the constant name.

• Convert selection to global constant
If a certain expression should be converted to a global constant, you can select it and provide the constant name. A global constant will be created within the current package, of a type that is determined from the expression. All occurrences of the expression in the current package will be replaced by the constant name.

• Replace assignment with initialization
If a local variable assignment is purely for initialization, you can move it to the declaration of the variable. You can right-click on a statement or select multiple statements.

If the results of a refactoring action are not satisfactory, you can simply undo it.

9.6 Code Contents layout
The layout of the Code Contents are changed so that variables, constants, types, and exceptions are now placed in a separate folder instead of the root of the tree view.

The Browse Back and Browse Forward buttons have been removed. Their functionality has been replaced by the global navigation function.
10. SQL Window Enhancements

The following enhancements have been made to the SQL Window.

10.1 Multiple statements

The SQL Window can now execute multiple statements simultaneously, and display all results on separate tab pages:

When switching between result tabs, the corresponding statement or PL/SQL Block will be highlighted in the SQL Editor. Statements that raised an error are marked with a red X on the corresponding tab. The error positions are marked with a red underscore in the SQL Editor.

10.2 Graph Window

You can create a Graph Window from the current result set by pressing the Graph button and following the wizard as described in chapter 3. To pre-select the data for the X and Y axis, you can select the columns in the grid. You can also select a range of cells or rows to limit the data.

Pressing the arrow button next to the Graph button will show a popup menu with several predefined styles and a From template item (see chapter 3). By selecting one of the items of this popup menu, a default graph will be displayed from appropriate columns of the result grid.
10.3 Alternate grid row colors

The SQL Window can now display alternate rows with alternate colors, so that long rows can easily be identified:

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7802</td>
<td>17-12-1986</td>
<td>600.00</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7499</td>
<td>ALLEN</td>
<td>SALESMAN</td>
<td>7688</td>
<td>20-04-1981</td>
<td>1600.00</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>7531</td>
<td>WARD</td>
<td>SALESMAN</td>
<td>7698</td>
<td>22-02-1981</td>
<td>1500.00</td>
<td>500.00</td>
<td></td>
</tr>
<tr>
<td>7586</td>
<td>JONES</td>
<td>MANAGER</td>
<td>7839</td>
<td>02-04-1981</td>
<td>2975.00</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7898</td>
<td>MARTIN</td>
<td>SALESMAN</td>
<td>7698</td>
<td>28-09-1981</td>
<td>1250.00</td>
<td>1400.00</td>
<td></td>
</tr>
<tr>
<td>7368</td>
<td>BLAKE</td>
<td>MANAGER</td>
<td>7839</td>
<td>01-05-1981</td>
<td>2650.00</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>7392</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>7839</td>
<td>09-06-1981</td>
<td>2450.00</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7039</td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>7666</td>
<td>19-04-1987</td>
<td>3600.00</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7498</td>
<td>KING</td>
<td>PRESIDENT</td>
<td>7666</td>
<td>17-11-1981</td>
<td>5000.00</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7844</td>
<td>TURNER</td>
<td>SALESMAN</td>
<td>7698</td>
<td>08-09-1981</td>
<td>1500.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>7876</td>
<td>ADAMS</td>
<td>CLERK</td>
<td>7708</td>
<td>23-05-1987</td>
<td>1100.00</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7901</td>
<td>JAMES</td>
<td>CLERK</td>
<td>7888</td>
<td>03-12-1981</td>
<td>950.00</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>7902</td>
<td>FORD</td>
<td>ANALYST</td>
<td>7666</td>
<td>03-12-1981</td>
<td>3000.00</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7394</td>
<td>MILLER</td>
<td>CLERK</td>
<td>7702</td>
<td>23-01-1982</td>
<td>1300.00</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

The alternate color can be enabled or disabled through a preference, and you can also set the actual color. The null value cell color preference will take precedence over the alternate row color to indicate null values.

10.4 Grid column totals

By right-clicking on a result set you can display a column total by selecting one of the functions of the corresponding submenu:
The total will be displayed in a bar below the grid:

![SQL Window output](image)

When editing the result set, the totals will dynamically be updated.

To clear a total, right-click on the column and select None from the Column Totals submenu.

### 10.5 Grid selection indicator

When selecting a range of cells, or when selecting rows or columns, the status bar will indicate the number of selected rows and columns (RxC).

### 10.6 Excel export

The Excel export function now adds the SQL Text on a second page of the excel sheet.
11. Command Window Enhancements

The Command Window now supports the _USER and _CONNECT_IDENTIFIER variables for SQL*Plus 10g compatibility.

The SET COLSEP command has been added to control the column separator. Use SET COLSEP OFF for an empty separator.

The COLUMN command has been extended to support the NEW_VALUE option, so that variable values can be set by executing a select statement.

The SET SPOOLDIRECTORY command can be used to specify the default location for SPOOL files. This only affects SPOOL commands where no absolute path is specified.
12. Session Window Enhancements

The following enhancements have been made to the Session Window.

12.1 Session Filters

You can define session filters to limit the sessions displayed and/or to define which columns you want to see and in which order. You can include/omit columns from the v$session table, or add additional column joined from other tables.

At the top of the Session Window you can select which filter you want to use:

In this case the All sessions filter is used. To view or edit the filters, press the Define Session Queries button on the toolbar:

Here you can define the Session Filters on the first tab page. The Caption will be displayed in the filter selection list. The Query will be executed to populate the session list.
12.2 **Concatenate hint**

To concatenate the value(s) of multiple rows of a session detail query, you can use the /* concatenate */ hint in the SQL text. For example:

```sql
select sql_text from v$sqltext_with_newlines
where address = hextoraw(:sql_address)
and hash_value = :sql_hash_value
order by piece
/* concatenate */
```

The `sql_text` column of all rows returned by this query will be concatenated, and displayed as one value in the session detail grid.

12.3 **CLOB and LONG columns**

CLOB and LONG columns are now also supported in the session details. When a CLOB or LONG value consists of multiple lines, the height of the corresponding row will automatically be increased to show all lines.
13. Editor Enhancements

The following enhancements have been made to the editors.

13.1 Global navigation

A global Navigate back and Navigate forward button have been added to the main toolbar:

Whenever you move to an absolute location within the same editor or in a different editor, a navigation bookmark is added to which you can return later. This way you can quickly find your way back if a function takes you to a different location, such as a find, hyperlink navigation, opening a new window, and so on.

13.2 Global bookmark list

The Bookmark list item of the Edit menu will bring up an global hierarchical display of all editor bookmarks:

In this tree view you can see the windows, editors, and lines where a bookmark is located. Clicking on a bookmark will bring the window and editor to the front, will navigate to that bookmark. You can dock the bookmark list to make it accessible at all times without blocking any other windows.

13.3 Color marks

You can add color marks in an editor to highlight specific sections. To do so, select the text you want to mark and press the Color marker button on the toolbar:

The background color of the selected text will change to the current marker color:

```sql
/* Fetch the name of the manager
select m.name into result
from emp e, emp m
where e.empno = xa
and m.empno = e.mgr;
```

To change the marker color, press the arrow next to the button and select a new color. To clear a color mark, place the cursor within the marked text without making a selection, and press the Color marker button.
13.4 Selection submenu
Right-clicking on a selection in an editor will now include a Selection submenu, with the same items as the Selection submenu from the Edit main menu.

13.5 Preferences
The Highlight color determines the color of all highlights such as parentheses, variables, and so on.
The Search hit color determines the color of the search hits found by the search bar.
14. Grid Enhancements

The following enhancements have been made to the grids.

14.1 Object Grid Filters

For all functions that display an object grid (export user objects, compare table data, and so on) you can define an object filter by double-clicking on a column and entering the filter criteria for that column. You can use wildcard characters to match multiple characters (% or *) or a single character (_ or ?). For example, double-clicking on the Name column of an object grid and entering EMP% will display all objects that start with EMP. The filter is not case-sensitive.

After defining a filter you can save it by right-clicking on the grid and selecting Save Object Filter from the popup menu. To reload the object filter later, right-click on a grid and select Load Object Filter.

14.2 Find function

The standard Find function now also works on grid data, in exactly the same way as in an editor. Just press Ctrl-F or select Find from the Edit menu when the grid has the focus.
15. Files > Format Preference Page

The following options are available on the Files > Format preference page:

The **Linebreaks** option controls how lines are terminated in text files. The can either be terminated in Windows style by CR/LF character pairs, or in Unix/Linux style by a single LF character. You can additionally specify that files are in Unix format on specific locations, and in Windows format otherwise. You can specify multiple directories on separate lines, and select a directory by pressing the Add location button (...). Note that not only files saved in this directory are formatted in Unix style, but also all files saved in subdirectories below this directory.

The **Export with smart quotes** controls how values are exported in CSV format. When disabled, all values are enclosed in quotes. When enabled, only those values that require them are enclosed in quotes. This are values that contain comma’s or quotes.
16. Active Query Builder Plug-In

PL/SQL Developer 7.0 introduces a new Query Builder Plug-In interface for 3rd party query builders. With the release of PL/SQL Developer 7.0 we have a Plug-In for Active Query Builder (http://www.activequerybuilder.com) from Active Database Software, Ltd:

If a Query Builder Plug-In is installed, then the standard Query Builder function will invoke the Plug-In. Right-clicking on the Query Builder button allows you to select a query builder from a list of all installed query builders. The most recently used query builder will be invoked by default when you subsequently use the Query Builder function again.
17. Oracle File System (OFS)

The OFS allows you to store all your files (sources, scripts, reports, and so on) in the Oracle database. This has the benefit that both the database objects as the files that operate on these objects can be stored in the same database, and can be accessed by all users that have access to this database. Furthermore, it is guaranteed that the database objects and files are consistent, and are backed up and recovered in a consistent way.

Files can be saved and opened in the usual way, with a standard file open/save dialog with some extensions.

17.1 OFS Manager

Before anybody can use the OFS from within PL/SQL Developer, you have to use the OFS Manager to define an OFS Location Directory with one or more OFS Locations. For each location you need to install the OFS database objects.

Creating an OFS Location Directory

To start the OFS Manager, go to PL/SQL Developer’s Preferences and select the Directories page. At the bottom of this page you can find the OFS Manager button. After starting it you get the following screen:

![OFS Manager Screen](image)

This is an empty OFS Location Directory, for which you can add locations, and which you can save in an OFS Location Directory file (.ldf). This file is used from within PL/SQL Developer to browse the directory and to connect to locations and browse the files.
Adding OFS Locations

To add an OFS Location, press the Add location (+) button. Now you can enter the following information:

- **Name** – Will be displayed in the location list in the file selector.
- **Database** – The database where the OFS files will be stored.
- **Schema** – The schema that holds the OFS database objects (tables, packages, and so on). This user requires the **resource** role privilege and the **query rewrite** system privilege to create its objects.
- **User name** – The name of the Oracle user that will be used to access the OFS. If you do not specify an OFS user name, then the user that accesses the OFS will need to specify a user name and password, or the OFS will be accessed as the current PL/SQL Developer Oracle user.
- **Password** – The password of the Oracle user that will be used to access the OFS. Only useful if the **User name** field is entered.

After creating one or more locations, you can save the OFS Location Directory by pressing the **Save** button. If you save the file in the PL/SQL Developer directory under the name **OFS.ldf**, then PL/SQL Developer will automatically pick-up this directory. Otherwise the PL/SQL Developer user will need to point to the correct OFS Location Directory file in the preferences.

Installing the OFS database objects

After adding a location to the directory, you need to install its database objects. If the OFS schema and (optional) OFS user do not yet exist in the database, you will first need to create them:

```
create user OFS identified by <password>;
grant connect, resource, query rewrite to OFS;
create user OFS_USER identified by <password>;
grant connect to OFS_USER;
```

Next you can install the OFS database objects by pressing the **Install database objects** button. After supplying the password of the schema user, the database objects will be created, and the location is ready for use.
17.2 OFS Usage

After creating an OFS Location Directory, installing the locations, and making the directory file available to PL/SQL Developer, you can use the OFS Save As item from the File menu to save a file into the OFS. The following dialog will appear:

You first need to open a location, after which you can save the file or create a directory:

To open a file from the OFS, you can use the OFS Open item from the File menu, or use the Reopen menu. OFS files have the path name OFS:\Location\Path (e.g. OFS:\Chicago\Programs\Employee.bdy). A previously opened OFS file can be saved by using the standard Save function. You can use the Save As function to save an OFS file to the standard file system. The OFS Save As function can be used to save it in the OFS under a different name or in a different directory.

When saving or opening a file, you can select the Keep locked option to lock it. Other users cannot lock or overwrite a file that is locked by you.
Right-clicking on a file in the file selector gives you the option to Cut, Copy, Paste, Delete or Rename the file. You can also view and change object properties from this popup menu. Properties you can change include the Read-only status, the Locked status, and the Compressed status. Files can only be compressed on Oracle10g and later, since it uses the `utl_compress` package which is only available since 10g.
18. Plug-In Interface

The following functions have been added to the Plug-In interface:

- CanClose
- PlugInName
- PlugInSubName
- PlugInShortName
- SYS_XPStyle
- SYS_TNSNAMES
- IDE_ShowDialog
- IDE_DebugLog
- IDE_GetParamString
- IDE_GetParamBool
- IDE_GetGeneralPref
- IDE_GetProcOverloadCount
- IDE_SelectProcOverloading
- IDE_GetSessionValue
- SQL_CheckConnection
- SQL_GetDBMSGetOutput
- SQL_SetVariable
- SQL_GetVariable
- SQL_ClearVariables

See the plugindoc.pdf file in the PlugInDoc subdirectory in the PL/SQL Developer installation directory for detailed information about these new functions.
19. Miscellaneous enhancements

- Windows XP theme support added
- Text Importer is now a normal document window instead of a dialog
- Data Generator now has a Create Table option
- DBMS_Metadata DDL/XML items added to Object popup menu for appropriate object types
- Popup menu added to To-Do List with Copy, Print and Export items
- Find Database Objects now includes Browser Filter as search criterion
- Options page added to Compare User Objects with new “Target schema” option
- Export User Objects now allows you to select a user and to include the owner prefix
- Substitution variable history recall added
- Explain Plan function now also removes bulk collect clause from select statement
- Pressing “Ctrl+” and “Ctrl-” in the Object Browser aligns the width with the contents
- Support for BINARY_FLOAT and BINARY DOUBLE data types added
- Don’t Show This Message Again options under preferences (User Interface > Options)
- Project “Add file(s)” added with support for non PL/SQL Developer files
- Empty Authorization table now allows all functions by default (instead of disable all)
- Installer now supports OverwriteFiles=TRUE/FALSE option for silent install
- Message boxes now have a popup menu with “Copy as text”, “Copy as bitmap” and “Print”
- Message boxes now have a scrollbar if the message text is too large