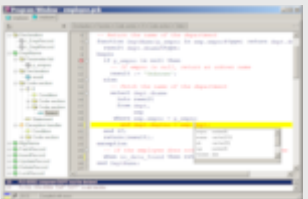


## The following are major features of PL/SQL Developer:

### Powerful PL/SQL Editor

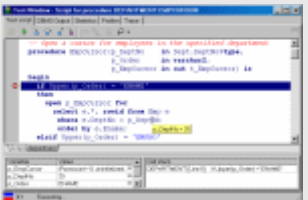
With its Syntax Highlighting, SQL and PL/SQL help, Object Descriptions, Code Assistant, Compiler Hints, Refactoring, PL/SQL Beautifier, Code Contents, Code Hierarchy, Code Folding, Hyperlink Navigation, Macro Library and many other sophisticated features, the Unicode compliant editor will even appeal to the most demanding user. Information is automatically presented to you when you need it, or is at most one click away.



The PL/SQL Editor

### Integrated Debugger

The integrated debugger offers all features you could wish for: Step In, Step Over, Step Out, Run Until Exception, Breakpoints, View & Set Variables, View Call Stack, and so on. You can debug any program unit without making any modification to it, including triggers and object types.



The Debugger

### PL/SQL Beautifier

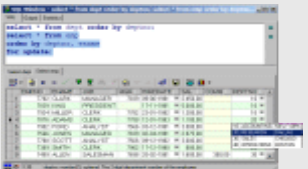
The PL/SQL Beautifier allows you to format your SQL and PL/SQL code through a user-defined set of rules. Your code can automatically be beautified when you compile, save, or open a file. This feature will increase your coding productivity and will improve the readability of your PL/SQL code if you are working in large project teams.



The PL/SQL Beautifier

### SQL Window

The SQL Window allows you to enter any SQL statement or multiple statements and view or edit the results in a grid. The result grid supports a Query By Example mode to search specific records in a result set. You can easily recall previously executed SQL statements from a history buffer. The SQL editor provides the same powerful features as the PL/SQL Editor.



The SQL Window

### Command Window

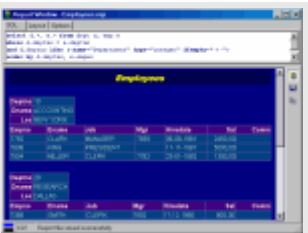
To develop and execute SQL scripts you can use PL/SQL Developer's Command Window. This window has the same look and feel as SQL\*Plus, and additionally has a built-in script editor with corresponding syntax highlighting. Now you can develop your scripts without the "edit script / save it / switch to SQL\*Plus / run script" cycle, and without leaving the comfortable PL/SQL Developer IDE.



The Command Window

### Reports

To run reports against your application data or against the Oracle dictionary, you can use PL/SQL Developer's built-in Report functionality. There are a number of standard reports, and you can easily create your own custom reports. These custom reports can be saved in a report file, which can in turn be included in the reports menu. This makes it very easy to run your own



The Report Window

frequently used custom reports.

You can use the [Query Reporter](#) freeware tool to run your reports without PL/SQL Developer, and to run them from the command-line.

### Diagrams

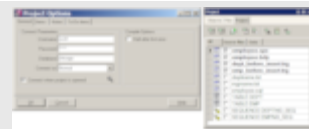
The Diagram Window allows you to create a graphical representation of a selection of objects. This way you can easily visualize (a part of) the database objects of your application or project and their relations. A diagram can be used for documentation purposes, but can also serve as a workspace. Right-clicking on an object provides access to all object functions, and double-clicking on an object will invoke the default action for the object type.



[The Diagram Window](#)

### Projects

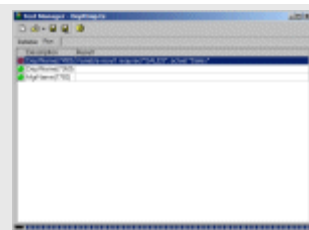
To organize your work you can use PL/SQL Developer's built-in project concept. A project consists of a collection of source files, database objects, notes, and options. It allows you to work within the scope of a specific set of items, instead of a complete database or schema. This makes it easy to find the project items you need, to compile all project items, or to move a project from one location or database to another.



[Projects](#)

### Regression Testing

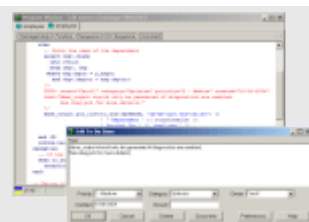
For regression testing you can use PL/SQL Developer's built-in Test Manager. You can define and run a Test Set, and quickly determine if all tests run correctly. In case of an error, the Test Manager can launch the [Debugger](#) to investigate the cause of the error. You can run Test Sets from the [Command Window](#) for automated regression testing.



[The Test Manager](#)

### To-Do Items

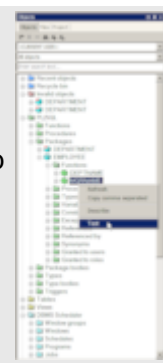
You can use To-Do Items in any SQL or PL/SQL source file to make a quick note that something needs to be done in this source file. You can access this information later from the To-Do List, either at the object level or at the project level.



[To-Do Items](#)

### Object Browser

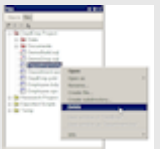
This configurable tree-view displays all information that is relevant to PL/SQL development. Use it to get object descriptions, to view object definitions, to create test scripts for the debugger, to enable and disable triggers and constraints, to recompile invalid objects, to query or edit tables or view data, to search for text in object sources, to drag and drop object names into an editor, and so on. The Object Browser additionally displays dependencies among objects, and lets you recursively expand these dependant objects (e.g. package references view, view references table, super/subtypes, and son on).



[The Object](#)

**File Browser**

If you frequently access source files and scripts from a limited set of locations (e.g. project directories, tool directories, and so on), you can use the File Browser for quick and easy access. Instead of opening a file selector to open or save files, you can access them directly from PL/SQL Developer's work area. From the File Browser tree you can access files and directories from user-defined locations on the local computer or on the network. You can quickly open, save, rename, and delete files here. You can use the File Browser as a docked or floating tool.

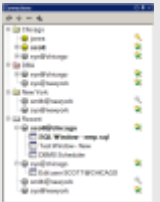


The File Browser

If you have the [Version Control Plug-In](#) installed, you can directly perform VCS operations from the File Browser, such as check-in, check-out, history, difference, and so on.

**Connection List**

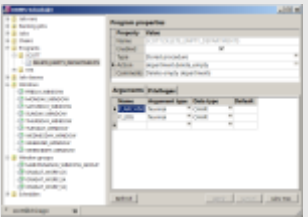
PL/SQL Developer allows you to work with multiple connections simultaneously. The dockable Connection List makes working with multiple connections easy with a hierarchical view where you can quickly open windows and tools for a connection, view which windows belong to a connection, and so on.



The Connection List

**DBMS Scheduler**

You can use the DBMS Scheduler tool to access the Oracle database scheduler (DBMS\_SCHEDULER) provided in Oracle10g and later. The tool can be used to manage DBMS Scheduler object definitions for object types such as Jobs, Programs, Windows, and so on. It can also be used to query job run information.



DBMS Scheduler

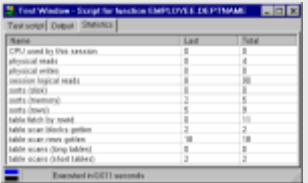
**Performance Optimizing**

To optimize the performance to your SQL and PL/SQL code, you can use the PL/SQL Profiler to view timing information of each executed line of PL/SQL code (Oracle8i and later).



PL/SQL Profiler

Furthermore you can automatically get statistics on executed SQL statements and PL/SQL programs. These statistics can include CPU usage, block I/O, record I/O, table scans, sorts, and so on.



Statistics

### HTML Manuals

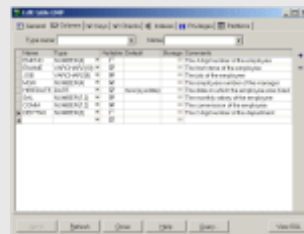
Oracle currently supplies on-line manuals in HTML format. You can integrate these manuals into PL/SQL Developer's environment to provide context sensitive help during editing, compilation errors, and runtime errors.



[HTML Manuals](#)

### Non-PL/SQL Objects

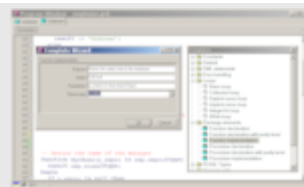
You can view, create and modify tables, sequences, synonyms, libraries, directories, jobs, queues, users and roles without using any SQL. Just enter the information in an easy to use form, and PL/SQL Developer will generate the appropriate SQL to create or alter the object.



[Non-PL/SQL Objects](#)

### Template List

PL/SQL Developer's Template List can be a real time-saver and can help you enforce standardization. Whenever you need to insert some standard SQL or PL/SQL code into an editor, and whenever you need to create a new program file from scratch, just click on the corresponding template.



[The Template List](#)

### Query Builder

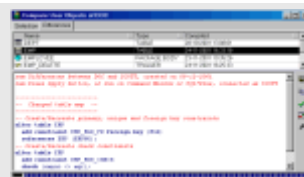
The graphical Query Builder makes it easy to create new select statements or modify existing ones. Just drag and drop tables and views, select columns for the field list, where clause and order by clause, join tables based on foreign key constraint definitions, and you're done. PL/SQL Developer's built-in Plug-In interface allows for 3rd party query builders, such as the [Active Query Builder](#).



[The Query Builder](#)

### Compare User Objects

After making changes to table definitions, views, program units, and so on, it may be useful to propagate these changes to another database user, or to find out what exactly the differences are. This may be another development environment, a testing environment, or a production environment. The Compare User Objects function allows you to compare a selection of objects, to visualize the differences, and to execute or save a SQL script that will apply the necessary changes.



[Compare User Objects](#)

### Export User Objects

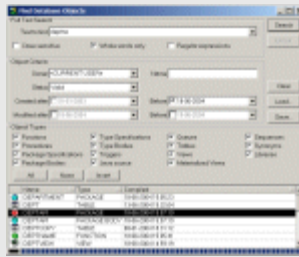
To export the DDL (Data Definition Language) statements of a selection of objects of a user, you can use the Export User Objects tool. This way you can easily recreate the objects for another user, or you can save the file(s) as a backup.



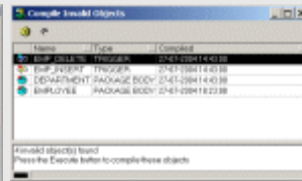
[Export User Objects](#)

### Tools

PL/SQL Developer includes several tools to make everyday development easier. You can recompile all invalid objects, search for text in database sources, import and export tables, generate test data, import text files or ODBC data, compare table data, monitor dbms\_alert and dbms\_pipe events, view session information, and so on.



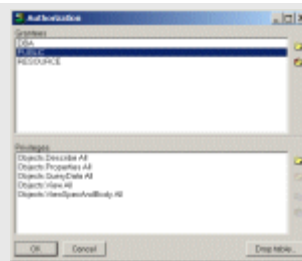
[Find Database Objects](#)



[Invalid Objects](#)

### Authorization

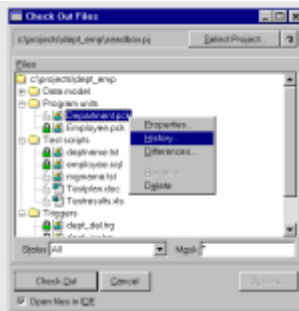
In most development environments you will not allow all PL/SQL Developer functionality in all databases. For example, in development databases you can grant all PL/SQL Developer functions, in test databases you can grant only the data querying/editing and object viewing functionality, and perhaps you do not want allow PL/SQL Developer to access your production databases at all. Through the PL/SQL Developer authorization function you can easily define which functions can be used by specific database users or roles.



[Authorization](#)

### Plug-In extensions

The functionality of PL/SQL Developer can be extended through Plug-Ins. We supply Plug-Ins on the [Plug-Ins](#) page that you can download without any extra costs. Plug-Ins can be provided by Allround Automations (such as the [Version Control Plug-In](#) or the [plsqldoc Plug-In](#)) or by other users. If you have a programming language that can create DLL's, you can even write your own Plug-Ins.



[Version Control Plug-In](#)



[plsqldoc Plug-In](#)

### Multi-threaded IDE

PL/SQL Developer is a multi-threaded IDE. This means that you can continue to work as SQL queries are executing, PL/SQL programs are executing, debug sessions are running, and so on. It also means that the IDE will not "hang" if you make a programming error: you can break execution or save your work at any time.

### Easy installation

No middle-ware required other than Oracle Net. No database object installation. Just a simple one-button installation procedure and you're ready to use it.