TOAD is a Windows-based graphical tool that allows for Oracle SQL and script execution.

This article will illustrate some of the more popular uses of TOAD. It will detail some of the keystroke saving features such as auto replace and code templates. There are several ways to obtain information and at least three ways to do most things.

For example, menu selection, hot-key sequence, right-mouse click, and assigned function key all have similar purposes. This article will focus on the three main TOAD utilities: the SQL Editor, the Schema Browser, and the Procedure Editor.

SQL Editor

Figure 1 illustrates the main TOAD interface, the SQL Editor. The SQL Editor allows you to easily enter and execute Oracle SQL or unnamed PL/SQL blocks. The results appear in the lower half of the screen.

F9 or the green arrow-shaped button allows you to run a single SQL statement or the SQL statement that the mouse cursor is currently on. This button also executes SQL that is high-
lighted from inside a PL/SQL routine. The output from this button appears in the DATA GRID tab in the lower area. There will also be an Explain Plan generated for this SQL, which you can view by clicking the EXPLAIN PLAN tab.

The button next to the green arrow-shaped button runs all of the SQL that is in the SQL area. The output from this execution appears in the SCRIPT OUTPUT tab.

Multiple connections to the same or different databases can be maintained. And, multiple TOAD windows can be open; the only limit is the amount of memory on your workstation! It is common practice to have connections to the test, development, and even production environments open at the same time. Under the Window menu item, it is easy to tile or cascade all open windows.

The clear button, encased in the blue circle in Figure 1, clears all the SQL in the window.

Pressing F8 brings up the TOAD SQL History or SQL Recall, as shown in Figure 2. This is one of the most popular features of TOAD because it makes it easy to bring up SQL worked on previously or, when tuning SQL 5 iterations ago. There are options to control how many SQL statements get saved. TOAD defaults to the last 100 SQL statements executed. Simply highlight the SQL and press the SQL button (encased in the blue circle in Figure 2) to load the SQL into the SQL Editor. This is also a dockable window; notice the push-pin button. This provides an auto-hide type function but still allows easy access to the SQL history.

ALT-Up Arrow and ALT-Down Arrow also access SQL. Recall from the SQL Editor window, and it will walk you up or down through recently submitted SQL.

There are a lot of TOAD features that can eliminate extra keystrokes when coding. One of the more popular features is the embedded alias processing. Using table aliases is a good practice to get into, and in TOAD, it greatly aids in the quick development of SQL statements.

As an example, starting with the Oracle SQL illustrated in Figure 3, add a SELECT and FROM clause and give each table in the FROM clause an alias.

Now, navigate to other parts of the SQL statement and enter the alias followed by a “.” and wait one second.

A column popup, as shown in Figure 4, will appear. Double click on any of the columns, and they will be pasted into your SQL session.
You can highlight several columns at once and hit return, and they will all be pasted into your SQL session as illustrated in Figure 5, complete with aliases! This saves a lot of typing and research, and the impending syntax errors because you did not spell the column name correctly.

This feature works anywhere in TOAD where you can work with SQL including SQL Editor and Procedure Editor.

The DESCRIBE window is another very popular feature. Highlight or just place the cursor on the object name in the FROM clause and press F4. This brings up a window shown in Figure 6, which shows the columns, data, scripts, indexes, and relationships. This window can be resized and moved around.

Column names can also be dragged and dropped from this window. Data can easily be viewed along with all the other information you see on these tabs. The INDEXES tab is useful when doing SQL tuning work. Being an old DBA, I am used to working with scripts. I like to see the script that can be used to recreate this object in the SCRIPTS tab.

Oracle data can be viewed, filtered and sorted, and saved in a number of formats, as will be discussed in the Schema Browser section of this article.

The Schema Browser shows all database objects the user owns or has access to.

Note the buttons along the top of Figure 6. You can easily start a wizard to edit an object, view key relationships, run statistics, or even truncate all of the data! But be careful out there!

Auto replace is typically used for typo corrections. Auto replace is activated in TOAD every time you hit the space bar. You can easily add your own items for replacement. Typically items for long table names, putline for PL/SQL coding, etc., are added.

Using the menu item Editor Options (Edit > Editor Options), as shown in Figure 7, you can see what is in AUTO REPLACE as shown in Figure 8.
Figure 6: F4 Describe Window

Figure 7: Accessing Auto Replace Options

Figure 8: Auto Replace Options
In this example, there are a variety of common typos that are addressed. You can also see the “pl” that I added for my `dbms_output.put_line` template. Every time I enter “pl” followed by the space bar, this code will be substituted in its place, as shown in Figure 9.

That was a brief overview of some of the top tips and techniques for using TOAD’s SQL Editor. But it was barely the tip of the iceberg! Topics not covered include generating template SQL code, saving data, code formatting, opening existing SQL, and saving SQL, which will be addressed in future articles.

**Schema Browser**

Pressing the Schema Browser button, as shown in Figure 10, accesses the Schema Browser. This interface shows all database objects the user owns or has access to.

On the left side, the logged in user appears. Using the dropdown menu can change this. TOAD can access any information on the database that the user has permissions to see.

The box just under the user box is a filter. You can have only certain objects appear in the tabbed navigator window (left side). This is handy when the application you are working with has hundreds of objects.

Notice again on the left side that the TABLES tab is displayed. There is a tab for every database object type. Tabs can be tiled with an option and right-mouse clicking on the tabs can hide tabs and uncheck the ones you do not want displayed.

The right side of the Schema Browser is remarkably similar to the F4 DESCRIBE screen in the SQL Editor. In fact, all of the same features discussed here also apply to the F4 DESCRIBE screen.

You can have a Schema Browser window open and drag and drop columns from this window back into the SQL Editor or the Procedure Editor.

Figure 9: Using Auto Replace

Figure 10: Schema Browser Interface
The DATA tab, as shown in Figure 11, has several options. You can add, delete, or edit the data using the buttons along the top, (encased in a blue circle on Figure 11). If you click on a row, then click the little book button (encased in a red circle on Figure 11), this will bring the whole row up for easy editing.

TOAD has two different reporting options.

1. You can print the displayed data.

2. You can save the displayed data in a number of formats. I say displayed data because you can use filters and sorts to get the data the way you want and the print and Save As options will only save the data displayed. In the SQL Editor, again, use the SQL itself to filter the data (WHERE clauses) and sort the data (ORDER BY clause).

Data can be saved in a variety of formats, as shown in Figure 12.

Code templates are similar to the auto replace except that they are signaled with the CTRL-space bar key combination.
There are several different options depending on what you selected in the upper radio group.

The INSERT statement selection allows you to put “COMMIT;” into the created text after a selected number of rows. Output options here include the clipboard; use it to paste back into the SQL Editor window, or a file. This feature along with DDL generation helps quickly and easily generate a test environment complete with test data.

**Procedure Editor**

The Procedure Editor is accessed via the Procedure Editor button. This brings up a procedure, function, package development environment, as shown in Figure 13, complete with symbolic debugger. Existing procedures, functions, triggers, and packages can be opened from a file or loaded into the Procedure Editor from the Schema Browser (PROCS tab, and either double clicking on the code or right mouse clicking and clicking on load into procedure editor).

All the coding shortcuts described in the SQL Editor also work here.

Code templates are similar to the auto replace except that they are signaled with the CTRL-space bar key combination. TOAD comes with a variety of existing templates.
and like auto replace, you can easily add your own.

Adding or changing the code templates is accessed from the menu item Code Templates (Edit > Editor Options > Code Templates). Figure 14 shows the code template options screen.

Figure 15 displays a sample cursor loop code template. This is accessed by entering “crloop” and then pressing the CTRL-space key combination. Note the vertical bar in the code: the cursor will be placed at that point. Also notice the substitution variables. TOAD will prompt you for these. In this example, you would enter in the cursor name.

The symbolic debugger supports break points, (stops the code), conditional break points (can stop the code after so many passes of a loop or when a variable meets a condition), watch variables, watch cursors, modify variable contents, and step in and out of called procedures. You can run to a break point, or you can step through the code one line at a time, or you can run the code until the debugger reaches the line with the cursor on it.

```
OPEN &c1;
LOOP
  FETCH &c1 | INTO :
  EXIT WHEN &c1%NOTFOUND;
END LOOP;
CLOSE &c1:
```

Figure 15: Cursor Loop Template
Another popular feature is that you can hover the cursor over any variable and TOAD will display its contents. If you hover over an implicit cursor, TOAD will display the current row’s information!

These buttons from the Debug menu bar, as shown in Figure 16, control the debug environment. Each of these items also has an associated button on the tool bar and as you can see, most of these items also have an associated function key.

**Summary**

We just scratched the surface of TOAD’s capabilities in this article. I have covered what I consider to be TOAD’s major features and features that will get you productive using TOAD.

Watch for additional TOAD articles on using the debugger and finding application problems.

---

Dan Hotka – Dan is a Training Specialist who has over 27 years in the computer industry and over 22 years of experience with Oracle products. He is an internationally recognized Oracle expert with Oracle experience dating back to the Oracle V4.0 days. Dan’s latest book is the Database Oracle10g Linux Administration by Oracle Press. He is also the author of Oracle9i Development by Example and Oracle8i from Scratch by Que and has co-authored six other popular books. He is frequently published in Oracle trade journals and regularly speaks at Oracle conferences and user groups around the world. Dan may be contacted at Dan.Hotka@ERPtips.com.