

# SUPERCHARGE YOUR PL/SQL DEVELOPMENT WITH THE RIGHT TOOL!

*Rumpi Gravenstein, Rumken Inc.  
Jen Croy, Rumken Inc.*

## TOPICS OF DISCUSSION

- Introduction
- Product Reviews
- Evaluation Metrics
- Product Comparison Matrix
- Conclusion

## INTRODUCTION

Over the years we have developed numerous PL/SQL triggers, procedures, functions, and packages. The development of this code had always been with the tool most readily available, in our case either the combination of SQL\*PLUS and a text editor or Oracle Procedure Builder. As we needed to write ever more PL/SQL code it became apparent that the right choice of a development tool would save us development effort and time. In fact, we now believe that the cost of most of these PL/SQL development tools can be quickly recovered with improvements in code development efficiency. We had intended to just search the literature for a product comparison of a number of PL/SQL development tools. However, when we could not find this comparison, we decided to do it ourselves. This paper performs that product comparison survey. The research in this paper was done without the knowledge or help of any of the product vendors reviewed and, as such, is an independent review of the products mentioned.

## PRODUCT REVIEWS

The product comparison was divided into several steps beginning with putting together a list of all of the products that we would include in our comparison. This review is limited to the products listed in Table 1.

Reviewed Product	Vendor	Reviewed Version	Web Site
SQL*Plus	Oracle	8.0.6	<a href="http://www.oracle.com">www.oracle.com</a>
Oracle Procedure Builder	Oracle	6.0.8.10	<a href="http://www.oracle.com">www.oracle.com</a>
PL/SQL Developer	Allround Automations	3.0.5.347	<a href="http://www.allroundautomations.nl">www.allroundautomations.nl</a>
RapidSQL	Embarcadero Technologies	5.6	<a href="http://www.embarcadero.com">www.embarcadero.com</a>
XPEDITER/SQL	Compuware/NuMega	3.50	<a href="http://www.numega.com">www.numega.com</a>
SQL Navigator	Quest Software	4.0c1	<a href="http://www.quest.com">www.quest.com</a>
TOAD	Quest Software	6.5.0.0	<a href="http://www.quest.com">www.quest.com</a> or <a href="http://www.toadsoft.com">www.toadsoft.com</a>
SQL-Programmer	Sylvian Faust International	2001.1002.0 Evaluation Millennium Edition	<a href="http://www.sfi-software.com">www.sfi-software.com</a>

Table 1. Reviewed Products

In the following sections we will provide our thumbnail overview of each product and what we see as its strong and weak points.

## **SQL\*PLUS**

We chose to use SQL\*Plus as a benchmark for all our product comparisons. We chose this environment as our benchmark because it is an essentially free environment (distributed with the database) that is installed by the database administrator, as part of the instance installation. When we use this environment for PL/SQL development, we use SQL\*Plus as a compilation tool and a text editor as the PL/SQL code generation tool. We use a separate text editor for PL/SQL development as the line editing support provided within SQL\*Plus is inferior to most freeware editors. A free text editor like PFE (Programmer's File Editor) includes standard functions like find and replace, automatic line numbering, and text indentation features that are not included with the line editor built into SQL\*Plus.

Over the course of time we developed a procedure in which we ended each trigger, package, function, or procedure with the SQL\*Plus SHOW ERRORS command. The SHOW ERRORS command will reveal the line number and column position of each error along with a plain text error message. Without this command SQL\*Plus will just return a rather cryptic message like "Warning: Package created with compilation errors". One last issue is that the given error line number is the PL/SQL block line number which does not necessarily agree with the file line number. Line number discrepancies can occur if the file contains SQL\*Plus commands or even blank lines prior to the start of the PL/SQL block.

The SQL\*Plus environment provides little help with tuning and debugging PL/SQL code. In this environment the developer must perform sophisticated tuning by either memorizing the required commands or consulting other Oracle manuals and then manually entering the commands to generate explain plans or tkprof data. High level statistics are available through the SQL\*Plus SET TIMING command. SQL\*Plus does not provide a simple way by which the developer can single step through the code, set break points, or examine or set intermediate data values. However, this type of debugging support can be found in all of the other tools we reviewed.

Finally, the PL/SQL development help system is not directly linked to the SQL\*Plus environment. Therefore context sensitive help is not available. When we needed to find information about a particular PL/SQL command, our only option was to start a separate process and then drill down to the desired Oracle PL/SQL reference documentation.

For the very experienced developer this tool set combination can be functional in situations where not a lot of work is being done. It is also "free" and, as we noted earlier, usually installed and configured by someone else, a plus if you are not planning on doing much work in the environment or do not have the freedom or time to install another product.

## **ORACLE PROCEDURE BUILDER**

We reviewed the Oracle Procedure Builder version that is distributed with the Internet Developer Suite. If you would like to purchase this product, visit the Oracle Store on the web. This environment is a significant step forward from the previously described SQL\*Plus environment. Among the many improvements is a GUI integrated development environment from which one can navigate through the Oracle database objects and then bring a desired PL/SQL object into a built in PL/SQL editor. This object navigation interface is, appropriately enough, referred to as the Navigator window.

The integrated PL/SQL editor includes many features including a compiler, an error locator which places the cursor on the offending line, and a syntax palette (Figure 1) which can be used to insert standard PL/SQL constructs directly into your code. This tool has support for the many language extensions that are native to the old Developer 6i product suite but not supported outside of that environment. These extensions include PL/SQL library support and a number of custom packages like text\_io, ole2, dde, and exec\_sql among others.

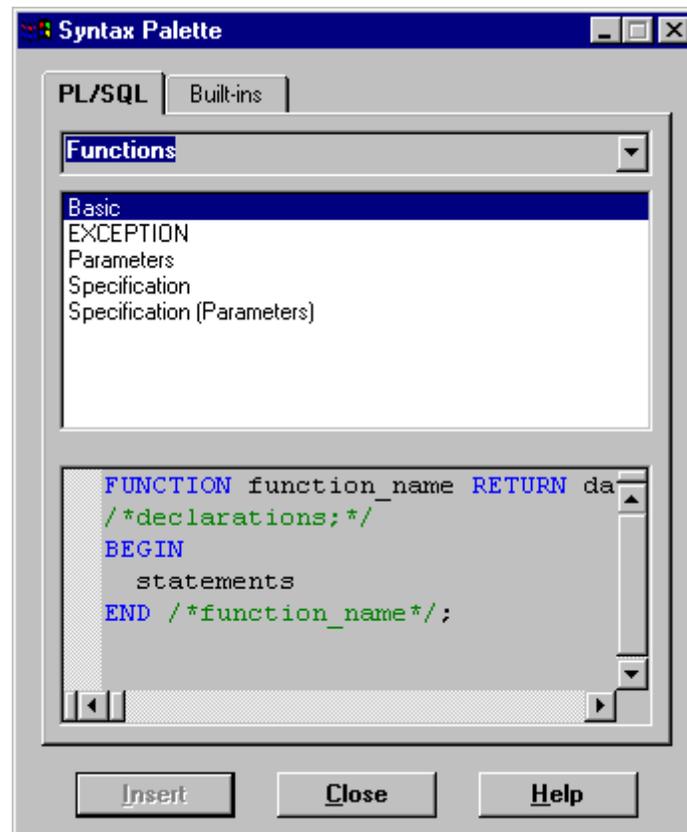


Figure 1 – Oracle Procedure Builder Syntax Palette

Oracle Procedure Builder also includes a sophisticated debugger which can be used to single step through code, break on a line, or with some coding, break on a loop iteration or break on a variable value via the use of a debug trigger. Data values can be examined and changed as the program is running, which is a very nice feature. The debugger interface can also display your PL/SQL module call stack (Figure 2). Also nice with this tool is its built in GUI trigger editor. The trigger editor provides a number of radio buttons to specify trigger options, removing the requirement to memorize the correct SQL trigger syntax.

This tool has a fairly nice help system. Most of the interface windows have a help option that will automatically display context sensitive help. Unfortunately, this level of help support is only available for aspects of the tool and not the underlying PL/SQL language. For help with the PL/SQL language, the user needs to find the Oracle PL/SQL documentation and then navigate to the desired area within that documentation set. This approach is no better than the one provided with our reference environment, SQL\*Plus.

The installation of this tool is very straight forward as you simply start the Oracle Installer and select Procedure Builder from the list of products that are distributed with Oracle Developer. This installation then turns into a several mouse click operation through the installer environment. A nice feature with this installer is that it will automatically install SQL\*Net, if needed. Other tools reviewed required that the SQL\*Net environment be installed prior to that package's installation. Also nice is that this product can be integrated with Intersolv's PVCS Version Manager and Rational Clear Case version control tools. Unfortunately this is the only third party tool extension that Oracle supports.

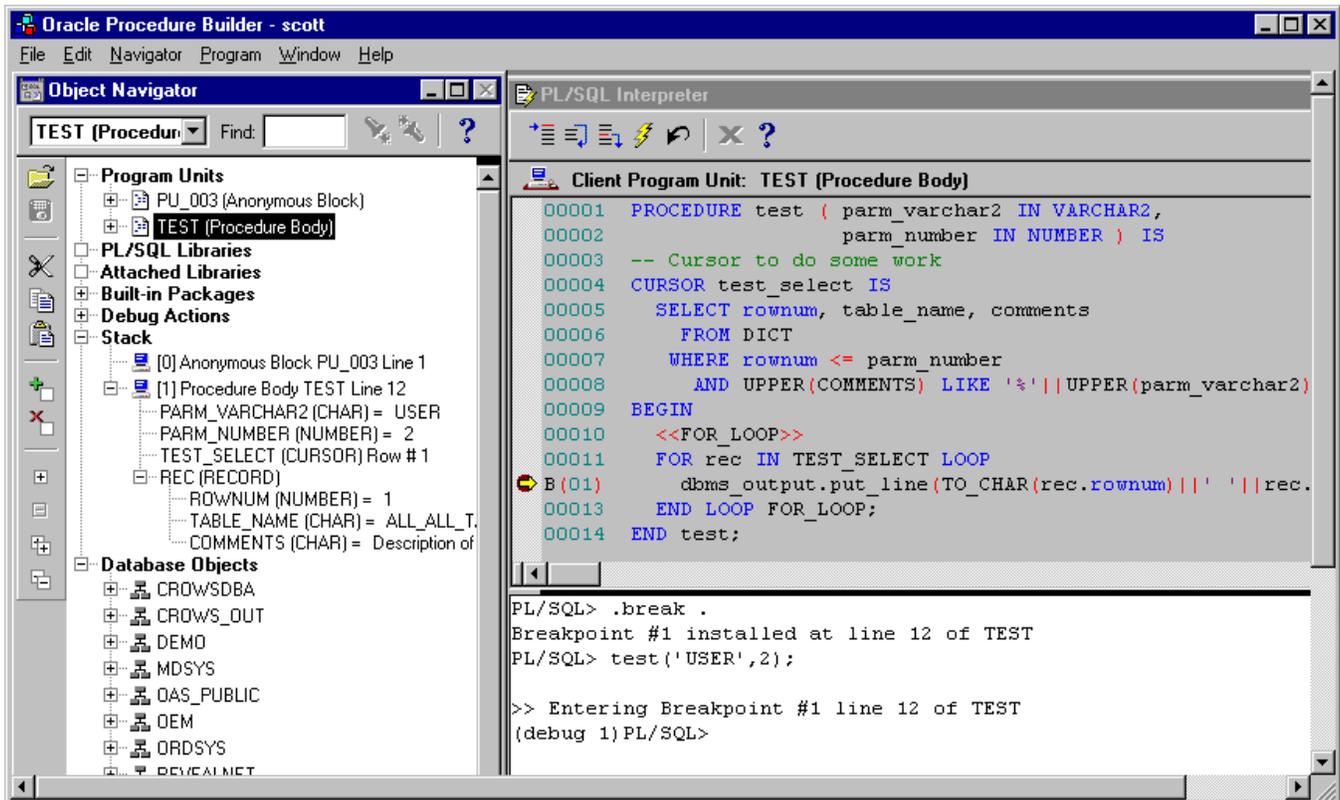


Figure 2 – Oracle Procedure Builder Debugger

A major negative for the Oracle Procedure Builder tool is that it can only be purchased as part of the larger IDS – Internet Developer Suite. Therefore, the product pricing (listed as \$4,995) includes costs for a number of other development environments (Oracle Forms, Oracle Reports, Designer, Oracle Portal, J Developer, ...) that may not be of interest. On the other hand, if you have already purchased or have need of the other components within the IDS tool set, this tool represents a nice bonus over our SQL\*Plus reference environment and should warrant your consideration during PL/SQL development efforts.

### PL/SQL DEVELOPER

PL/SQL Developer, an Allround Automations product, is developed by a company from the Netherlands and distributed through their web site at <http://www.allroundautomations.nl>. We were very impressed with the depth and breadth of this product. The nice experience began with the price, a nominal \$150 for a single user license, continued through the installation of the product, and right into its use. The installation simply required running the downloaded executable. The first time we entered the application we found that its navigation was straight forward with an easy capability to browse the objects for any schema. An additional feature is that the object browser includes a complete description of the viewed object (Figure 3). For instance, the interface provides a view of a table's columns, primary keys, unique keys, foreign keys, check constraints, triggers, foreign key references, referenced by (who is referencing this table), synonyms, granted to users, and finally granted to roles. As you might expect, each of these areas provides a very complete description.

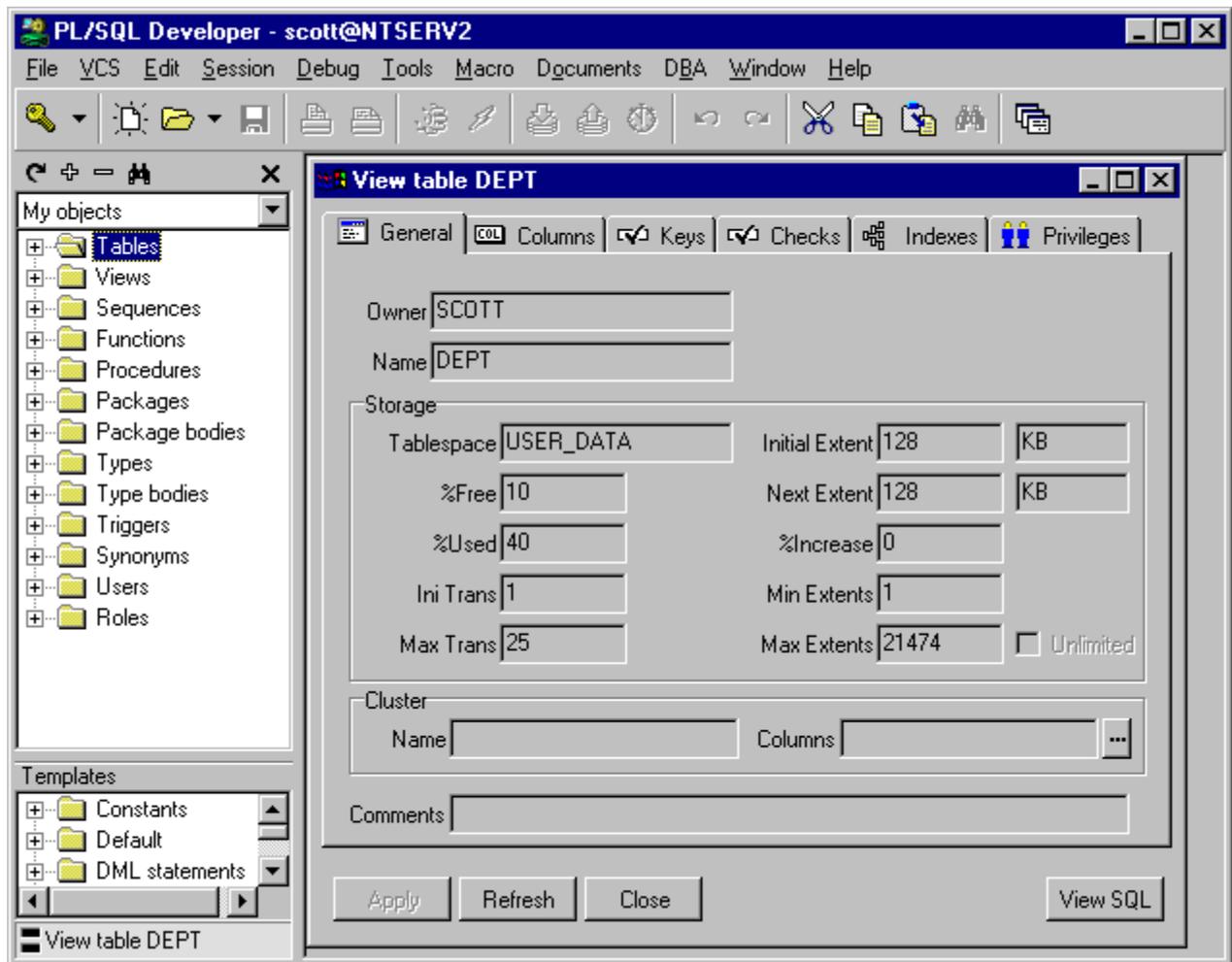


Figure 3 – PL/SQL Developer Object Browser

For PL/SQL objects the detail is similar. The PL/SQL object describe reveals the block parameters and the properties that you would find in the DBA\_OBJECTS view. While viewing a package, the editor provides a quick index that can be used to navigate to the start of any of the programming units in the source. The environment also provides some very nice built in features, like a code assistant which will automatically kick in to help identify parameters to standard functions and procedures. This feature taken along with the right mouse click menu that provides ‘on the fly’ object definitions adds a bell and whistle not found in our reference SQL\*Plus tool. The editor also includes a number of nice formatting features which you can customize through a preference dialog, like automatic code indenting and syntax highlighting with font colors to identify keywords, comments, strings, numbers, symbols, and custom keywords.

The PL/SQL help environment within this tool is superior once it has been setup. This tool incorporates the help files provided by Oracle through a slick indexing approach. Once you have gone through the steps of including the Oracle html manuals into the PL/SQL Developer, the help facility will immediately navigate you to the section that corresponds to the word or phrase that is being edited (Figure 4). This is a significant improvement over the help provided with the SQL\*Plus environment. Unfortunately, there is no context sensitive help available for the many features of the PL/SQL Developer tool. To find help on the tool, we had to refer back to the PL/SQL Developer manual. The program manual surprised us as it is a very well written description of the program’s operations, something we were not expecting from a company based in the Netherlands. Perhaps the folks at Allround Automations can look at adding context sensitive screen help with their next release.

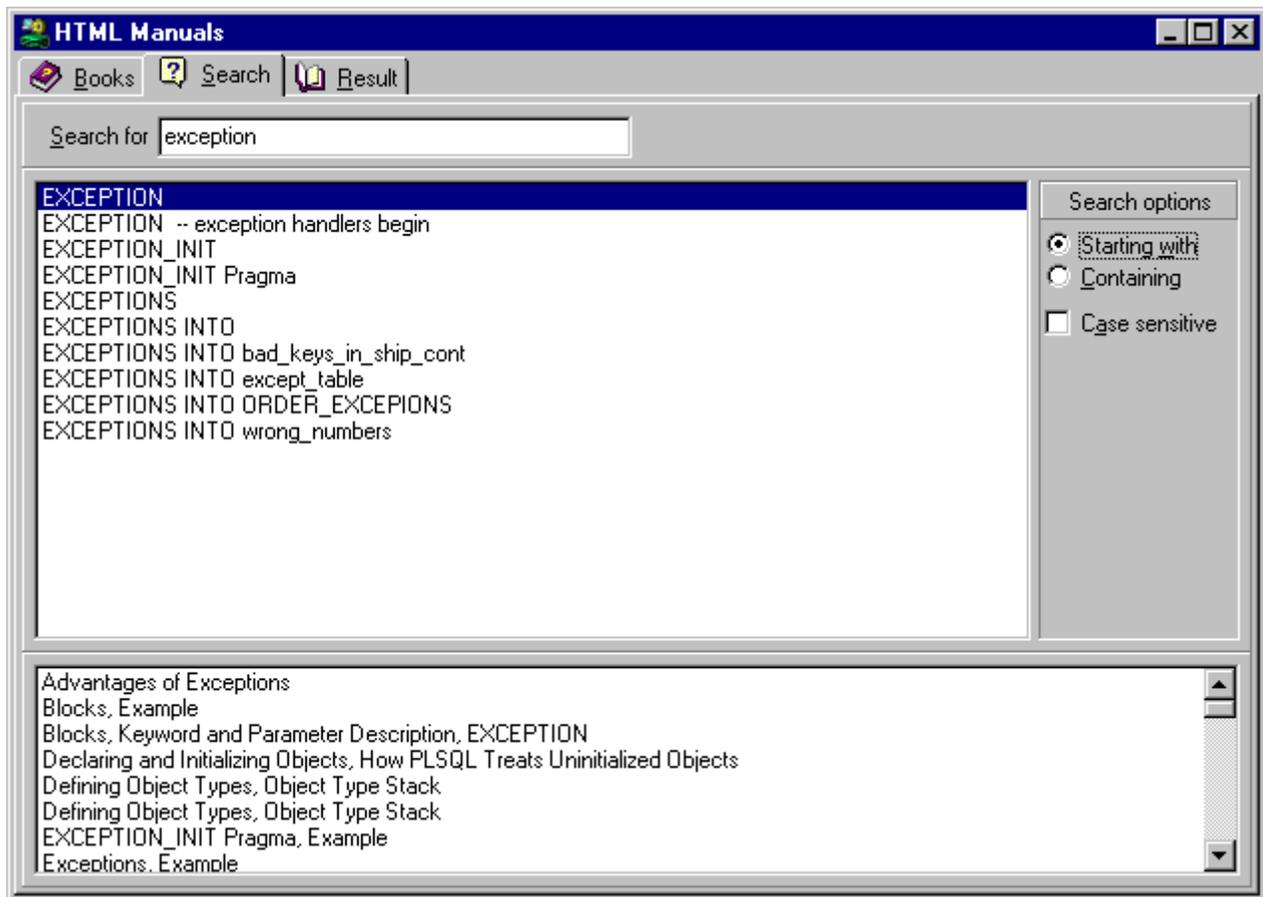


Figure 4 – PL/SQL Developer PL/SQL help facility

The PL/SQL Developer debugger is a very nice debugger (Figure 5). It has all of the standard features, like step in, step over, break, break on iteration, watch, change variable value, view the call stack, and set bookmarks. The tool will also automatically route DBMS\_OUTPUT text to the Output tab without requiring any configuration, the result we would expect from a development and testing environment.

This product has a number of ease of use features that we found very nice. The list starts with the login history screen. This screen retains a list of all users and instances which have been used to login to Oracle databases. Another nice feature is the built in SQL tuning support, with access to the explain plan and tkprof functionality. Perhaps the best ease of use indicator is the preferences dialog which contains 8 tabs. Each of these contain a number of options. The options range from the length of the login history list to which extensions should be used to store different type of program elements to hot key configurations. This is a very flexible tool.

The tool also includes an interface that you and third party developers can use to extend the product. There are already a number of these plug-ins available, some of which are currently being distributed free of charge by Allround Automations. See the vendor website for more details. We installed the Version Control Interface, one of the free plug-ins, which provides a native interface to all Microsoft SCC compliant version control products. SCC compliant version control software products include Microsoft's Source Safe, PVCS, Mortice Kern Systems MKS Source Integrity, and Clear Case. Other available plug-ins include tools that provide customized syntax highlighting, PL/SQL formatting, and a PL/SQL Wrapper utility.

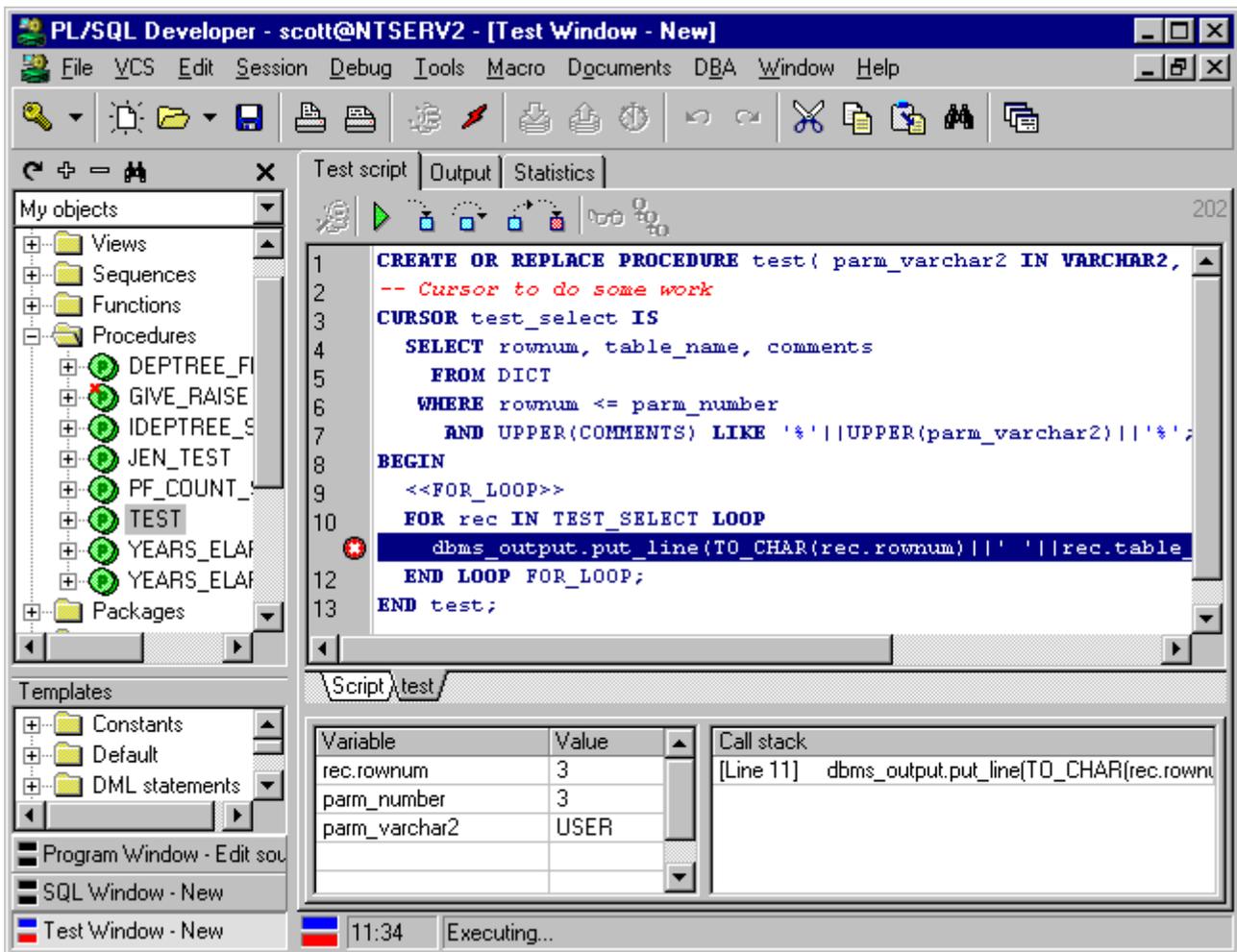


Figure 5 – PL/SQL Developer Debugger

## RAPIDSQL

RapidSQL was quoted to us at a single user price of \$895 for the base module and \$395 for the debugger. This application cannot be purchased directly from the web, rather you need to work with one of the Embarcadero sales representatives to get a price and then purchase the product from them with a purchase order. RapidSQL has a number of nice features. The RapidSQL user interface has an appealing look. As with most of the other applications that we reviewed, you start with a GUI navigator that browses through the database objects (Figure 6). It is easy to navigate to a schema and then from the schema through all the objects owned by that schema. This browser supports server side libraries, an object not supported by either PL/SQL developer or Oracle Procedure Builder navigators. Once you have identified a table the interface provides an easy method by which the data in the table can be viewed. Editing of table, view, sequence, and other object definitions can be also be done from the interface. We did not find a direct way in which to edit data within a table, however.

Other interface features include support for multiple workspaces. These workspaces can be used to quickly switch between different work areas. We quickly noticed the color coding of the different PL/SQL syntax elements. Also, template support is available for standard SQL statements (Figure 7). Support for PL/SQL appears to have been limited to three conditional statements: the IF statement, the EXCEPTION logic, and the WHILE loop. This is minimal PL/SQL template support to say the least. When it's time to optimize the PL/SQL code you have written, RapidSQL provides an interface to the explain plan functionality and displays an execution tree for use in optimizing your SQL statements.

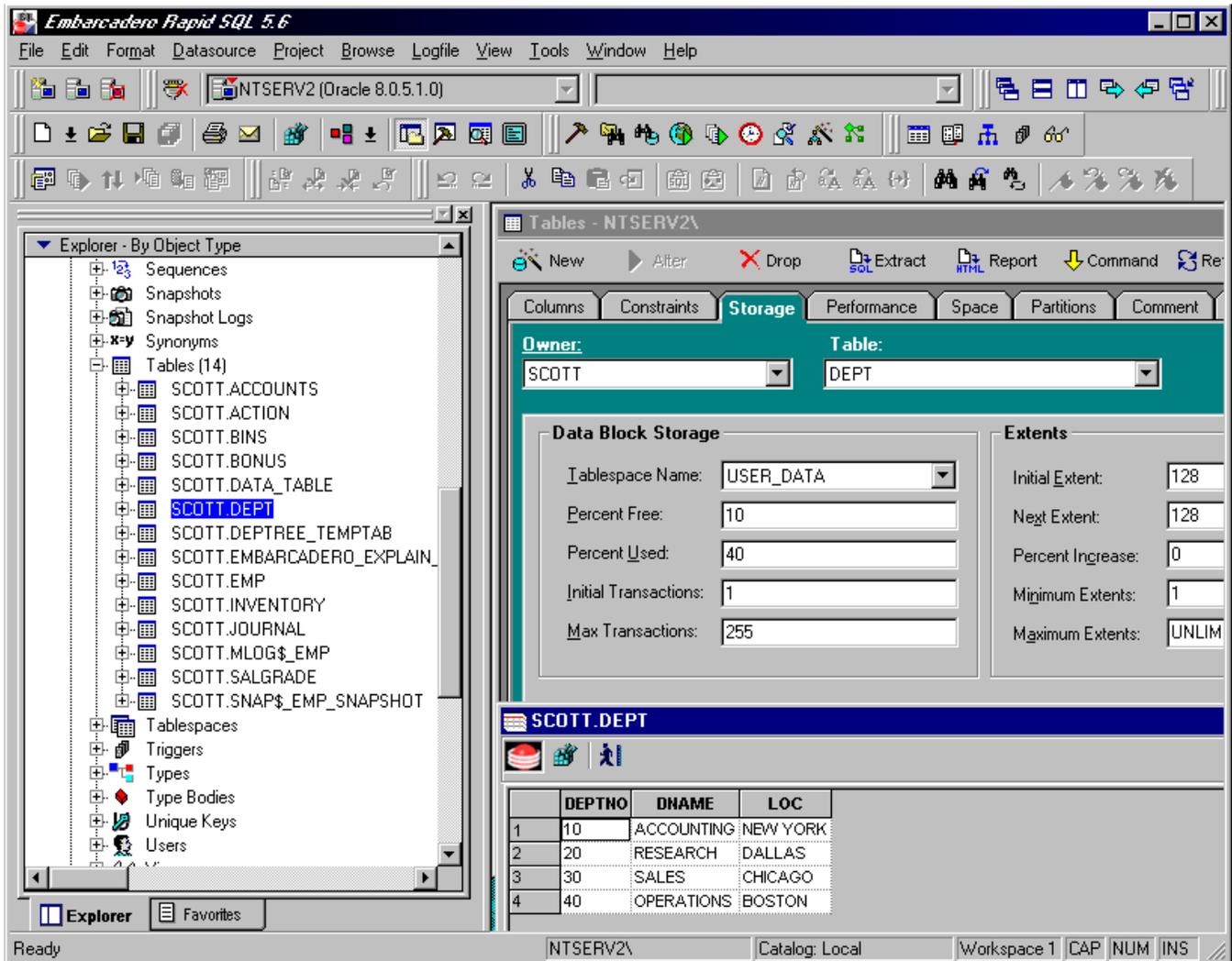


Figure 6 – RapidSQL Object Browser

Each window within the product has a help page associated with it. The help is usually very informative. However, there are exceptions. The tutorial on the integrated PL/SQL debugger is comprised of 5 or 6 screens. Yet, there is no clear method to get from one screen to the next within the tutorial.

The debugging environment has standard functionality like single step, step into, run to cursor, and a breakpoint option (Figure 8). But there is also room for improvement. We would recommend adding the ability to set a break point based on a variable value or a loop iteration. Another problem is that the RapidSQL tool does not have help support for the PL/SQL language syntax. To get PL/SQL help, we needed to find the PL/SQL documentation provided by the RDBMS installation, as was also needed with Oracle Procedure Builder and SQL\*Plus.

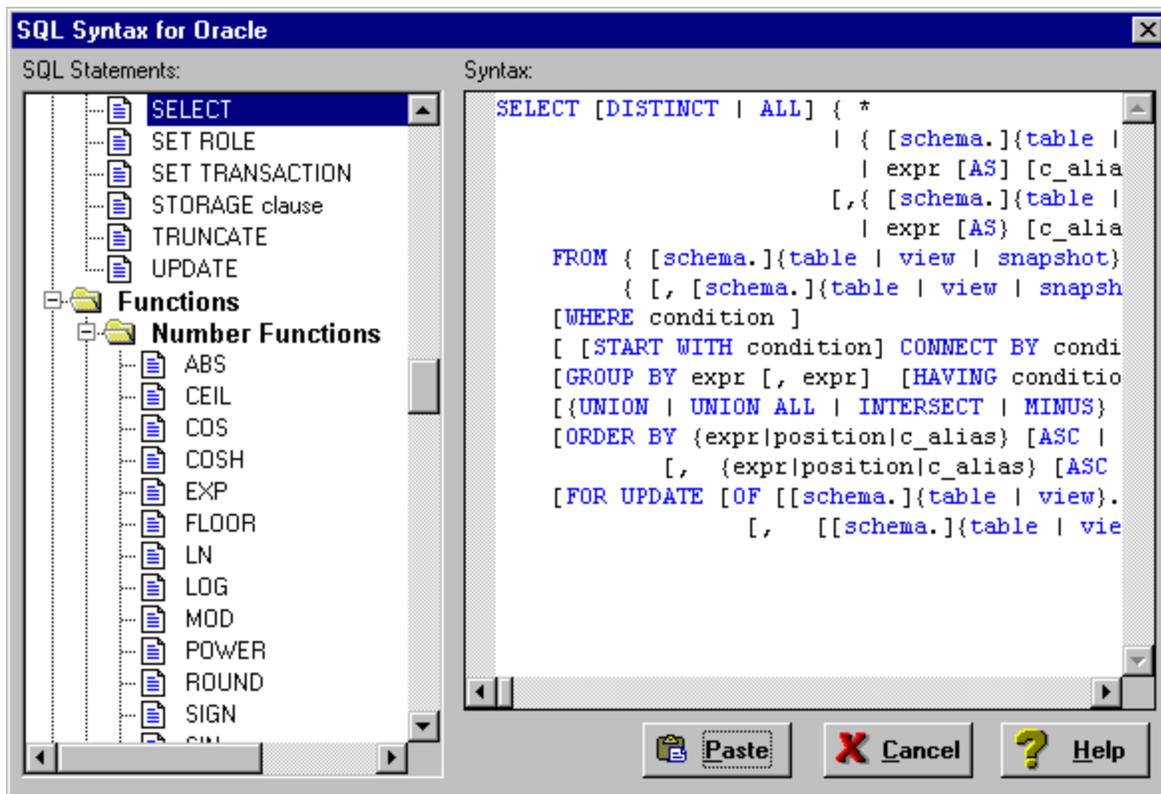


Figure 7 – RapidSQL Syntax Template

RapidSQL supports not only PL/SQL development but also Java and html development. Java and html development support is beyond the scope of this review. However, the support for these areas allows developers who have to work in these environments the possibility for a simpler, single development or maintenance tool. Given the increasing emphasis that Oracle is putting on Java development, this is a definite bonus.

The RapidSQL tool includes an interface to a version control tool. Tools supported with this product include: Source Safe, PVCS, and MKS Source Integrity. The interface to a number of different project management products is just one instance of the general nature of this tool. In addition to Oracle, RapidSQL also supports DB2, Sybase, and Microsoft SQL Server databases. Thus if you work in a heterogeneous environment with support or development responsibilities for Oracle and other databases, this tool may be a good choice. One area that shows the benefit of this cross database vendor platform support is the text search feature. RapidSQL gives you the capability to search across multiple database instances for a text string.

The customization possibilities within the RapidSQL tool are tremendous. There are 16 tabs that can be used to customize virtually every aspect of the environment to your personal liking. We found that this is one of the nicest aspects of the tool and can make you quickly feel right at home within the environment.

We conclude our review of the RapidSQL tool noting that the tool supports a number of add-on features outside of the direct path of PL/SQL development. For instance, there is a macro environment for automation of repetitive tasks, such as creating a project build or release from the version control tool. There is also a batch job environment, a separate product from Embarcadero, that can be used to schedule database backups and perform other operating system level functions. The tool also offers e-mail connectivity so that developers can either e-mail the results of output or be notified by e-mail upon the completion of some tasks. However, this tool does not offer an open interface by which third party developers can extend the basic functionality of the underlying product. Too bad.

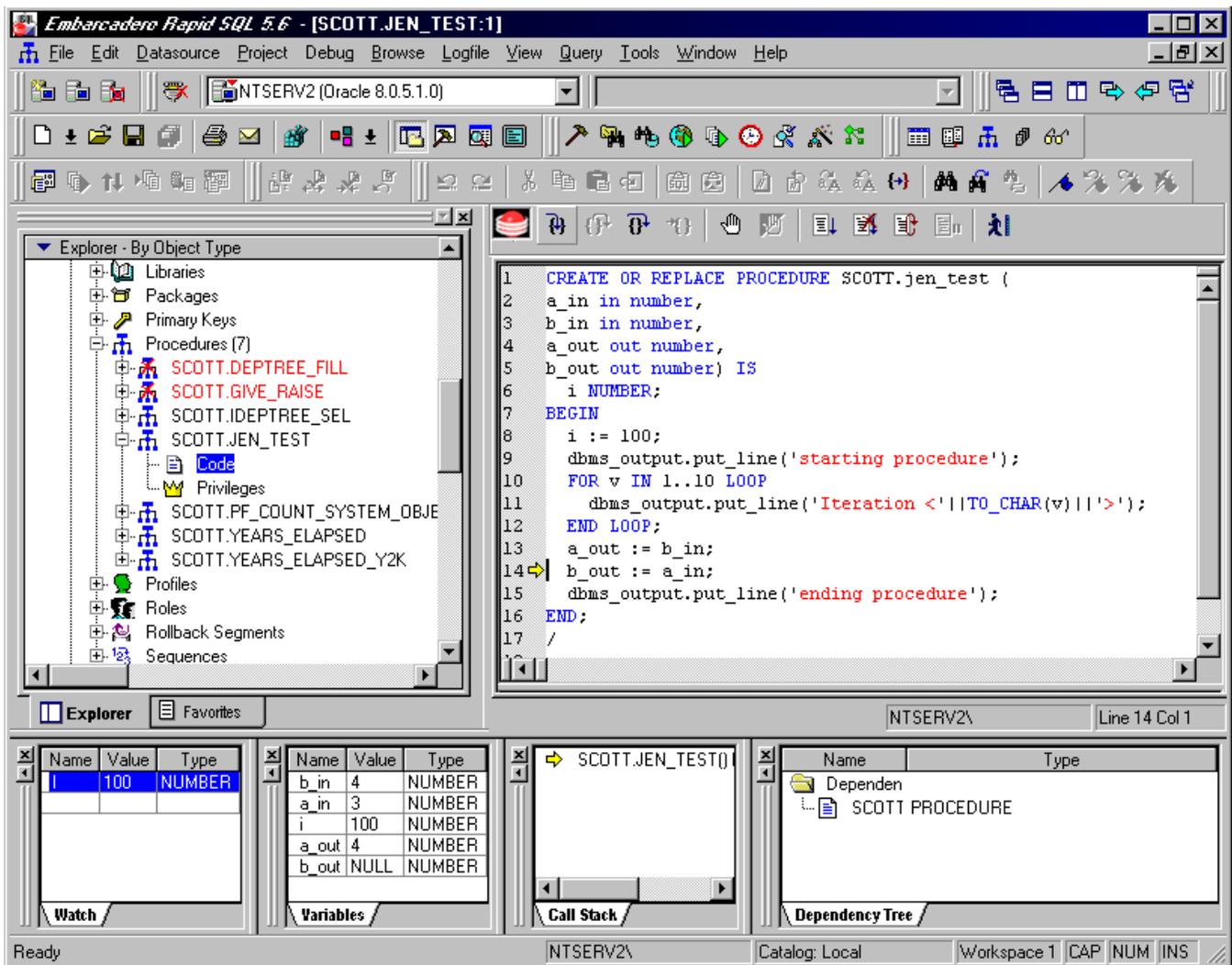


Figure 8 – RapidSQL Debugger

## EXPEDITER/SQL

We had planned on performing a full review of this product. Unfortunately we ran into a number of problems. First, Numega, the Compuware subsidiary now responsible for the product, does not provide a web site from which a trial copy can be downloaded. All of the other products we reviewed provided an internet accessible trial copy. After calling around, we were able to get a Numega Compuware representative to send us a trial copy which could be purchased for \$1800.

Unfortunately our problems did not end here. After we received the software and installed it on our test machine we noticed that the browser response, especially when viewing data, would sometimes slow down to a crawl. Another call to Numega technical support revealed a known problem in the version that we were reviewing. The Numega technical support representative provided us with an ftp download site from which we were to download a new, patched version. Unfortunately a security issue kept us from downloading the patched version from that site. After calling the Numega representative back with our problem, he suggested another alternative, a hand patch of the product. At that point, we stopped in our review efforts. Our impression is that there are too many issues with simply getting this version of the product installed to warrant a full review. If you are interested in this product we would recommend that you wait for the next maintenance release, which should be out soon. By the way, Expediter/SQL does appear to offer some very nice features like a straightforward user interface, the capability of debugging Java stored program units within the Oracle JVM, and a nice tutorial.

## SQL NAVIGATOR

We downloaded a full version of SQL Navigator (\$670) with the optional PL/SQL Debugger (\$520), Xpert Tuning (\$1350), PL/Formatter (\$125), and the Active PL/SQL Knowledge base (\$390). This represents quite an array of options that we believe can help in our development of PL/SQL code. The PL/Formatter and the Active PL/SQL Knowledge base are actually developed by another company, RevealNet. After we got the products installed and started running the SQL Navigator, we were pleasantly surprised with an automatic hint feature that made the initial introduction to the product go very smoothly. In fact this tool has a very nice and tightly integrated help system that explained virtually every screen. When this help system is combined with the Active PL/SQL Knowledge base and the help it offers with PL/SQL, we found this SQL Navigator to have the best help environment of all of the products we reviewed.

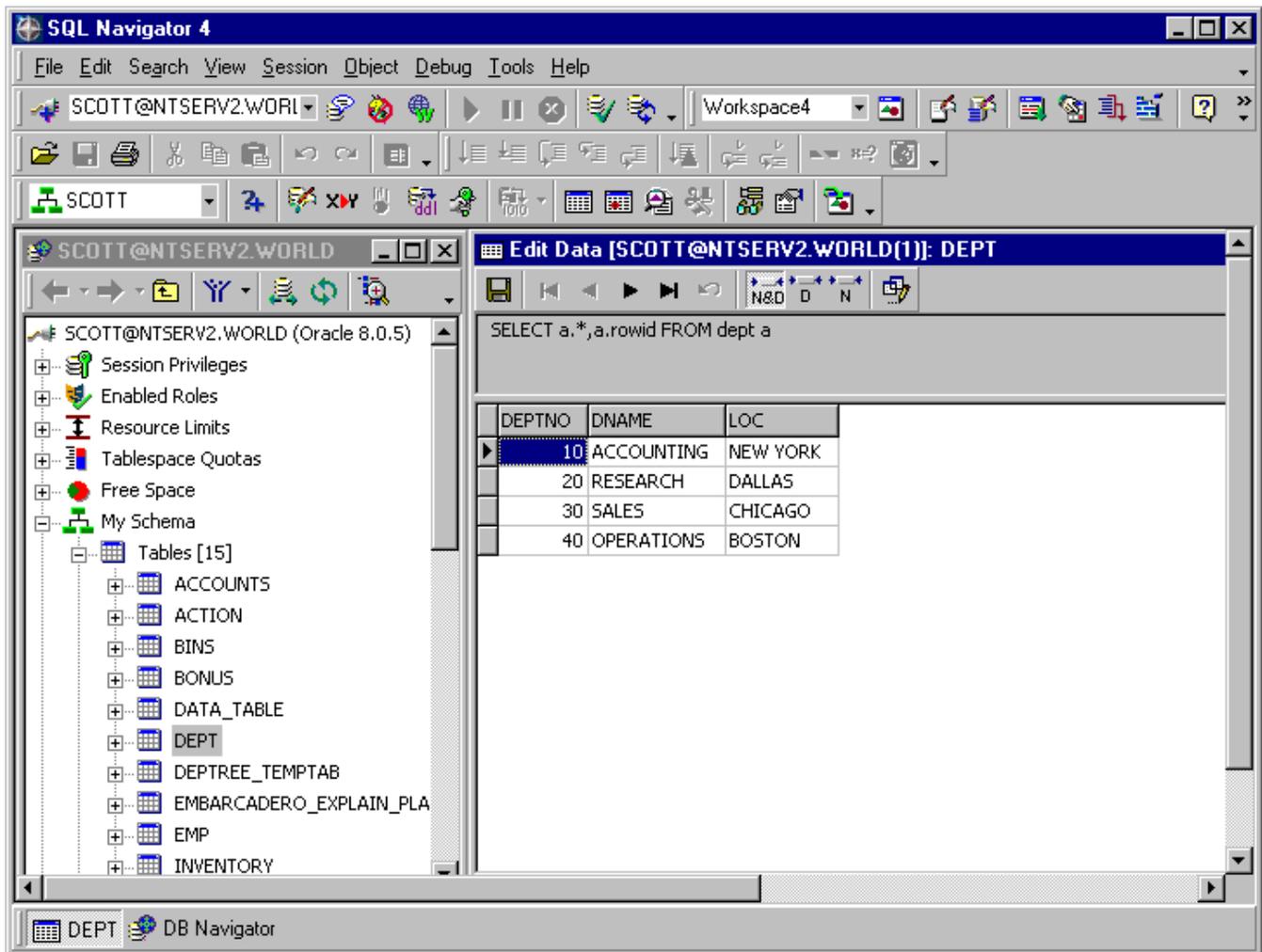


Figure 9 – SQL Navigator Object Browser

As in the other products we reviewed the SQL Navigator includes a navigation window, which can be used to navigate and drill down through database objects of all types (Figure 9). This navigation window is comparable to windows reviewed in other products. What extends this environment out a little beyond some of the other products is the availability of 4 workspaces to which you can add shortcuts to favorite objects found in the navigator. Then later when it's time to login with a test user account into the test instance, we were able to go back to the workspace and retrieve our saved login connection, short circuiting what would otherwise have been a slightly more difficult task. This mechanism then allows you to keep the

equivalent of a login history of only those connections for which you have an interest. If you like, you can use the navigation window to drill down further to browse and even edit data in a table.

This version of SQL Navigator does not include an interface to a version control system. This is very interesting as the documentation suggests that the prior version had included such an interface and that a future next release would again include a version control interface. Too bad.

The optional PL/SQL debugger includes many of the standard debugger features: Continue, Run to Cursor, Trace Into, Trace out of (Figure 10). The debugger can, with some coding, be used to debug “external” procedures. External procedures are database procedures that are being called by an external application, such as a custom client server application written in VB, Delphi, Power Builder, or Oracle Developer. The debugger can also be made to stop on an iteration with the setting of a breakpoint condition. A very nice feature.

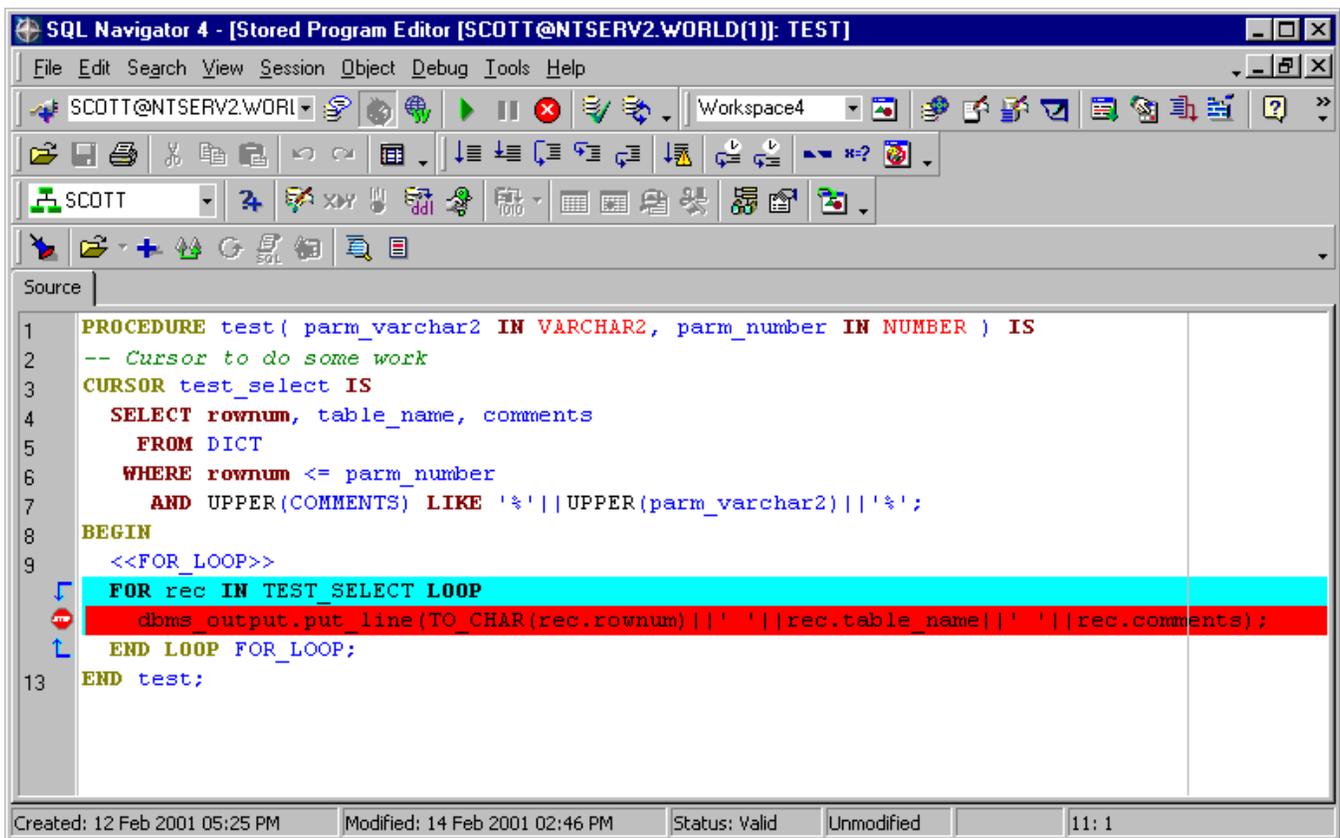


Figure 10 – SQL Navigator Debugger

The PL/SQL editor includes a nice set of prompts to quickly reveal the begin/end pairs that make up the current block of code in which the cursor is positioned. This feature can be very helpful in understanding more complex code segments with multiple block levels. The debugger must be activated with a separate button press. We found this to be slightly annoying as several features change in the debugger mode. For instance, the line error location feature is turned off while in debug mode. Therefore, while the debugger is turned on the error display is limited to an Oracle error along with the somewhat cryptic error location like line 11 column 5. When not in debug mode the problem line is highlighted and the error is displayed in an output pane immediately below the SQL editor pane. Although switching between debug and non-debug mode is a trivial task, we still wonder why the debug mode is not turned on automatically when the step into debug function is run.

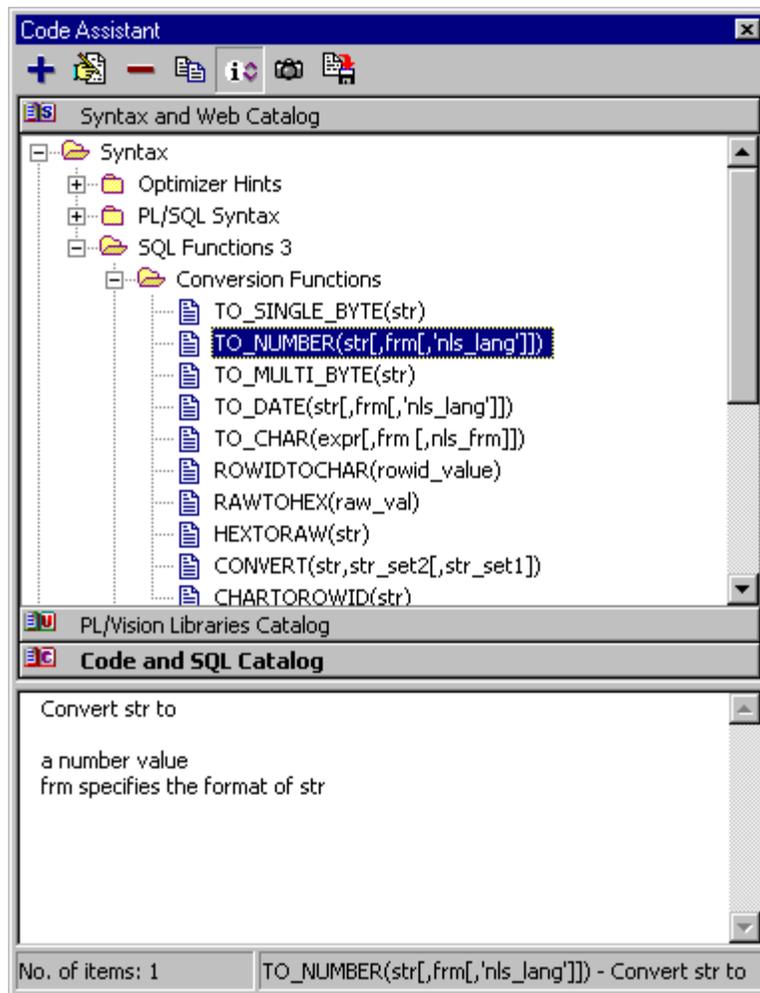


Figure 11 – SQL Navigator Code Assistant

We feel the PL/SQL template support within SQL Navigator is excellent (Figure 11). The PL/SQL templates are complete and extensible in that you can easily add additional code snippets into this template library. With the software we found templates for: Optimization hints, PL/SQL syntax, SQL Functions, Web Server PL/SQL Procedures, and PL/Vision Libraries. What we expected, but didn't find is support for standard SQL statements like SELECT, INSERT, DELETE and UPDATE.

We have a few final observations. This tool is feature rich with numerous customization options. In fact there might be too many options as the user interface is very busy. Second, the SQL Navigator tool does not include a documented open third party interface. Thus the product can only be extended by the developers at Quest Software and not by our or your in-house staff.

## TOAD

TOAD (\$670 for a single-user license), also released by Quest Software, has many of the same features as SQL Navigator. The PL/SQL Debugger (\$520), Xpert Tuning (\$1350), PL/Formatter (\$125), and the Active PL/SQL Knowledge base (\$390) are all also available for use with TOAD.

It appears that the only system requirement is 32-bit SQL\*Net. Installation wasn't quite as easy as with some of the other products. After downloading and unzipping a .zip file from Quest's web site, we had to run 7 separate setup programs, as we chose to install all of the optional add-ons. During the course of each setup program, we had to enter a separate authorization code for each of these programs.

Once installed, we launched the TOAD product. At first, we found the TOAD environment somewhat confusing. After a little familiarization and adjustment, we were up and running. The primary features most developers will want to use are the Schema Browser, the SQL Editor, and the Procedure Editor. These features can be accessed with the toolbar or the appropriate menu item under the “Database” menu. Each of these features will launch a new MDI window. The Schema Browser provides a hierarchical display of all database objects. The SQL Editor provides a window where SQL statements are executed. And PL/SQL programs can be edited within the Procedure Editor.

TOAD will not permit you to open a file unless you are already connected to a database. This can be very annoying, especially for developers who frequently do off-line editing, such as on a laptop or at home.

In the Procedure Editor, each opened file will appear as a new tab in the current Procedure Editor window (Figure 12). Note that a database connection must be active and Procedure Editor must be activated in order to open a file on the file system. Users should also be aware that clicking the “X” (Close) button for the Procedure Editor window will close all files in that Procedure Editor window. To close an individual file only, the user should right-click on the editor portion of the window. This will bring up the context menu, at which time the user should choose "Close File/Tab".

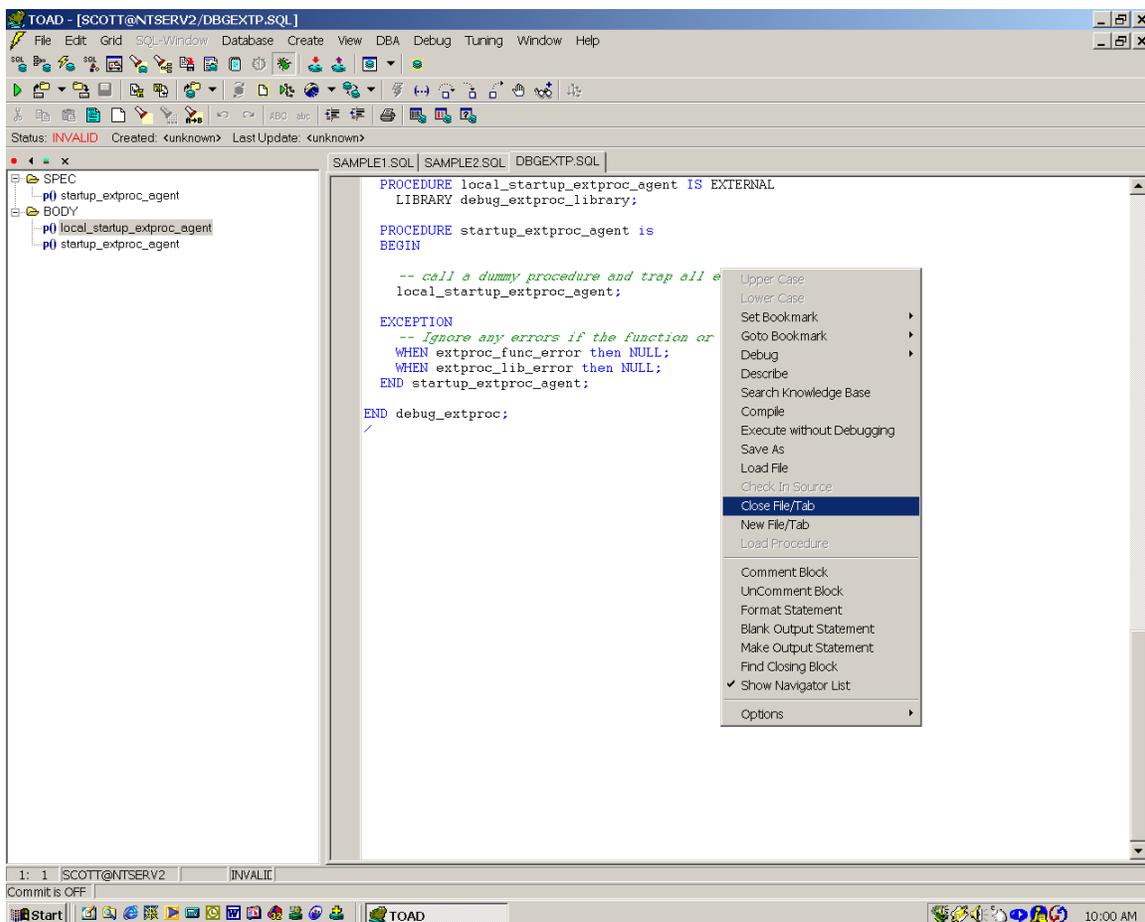


Figure 12 – TOAD Procedure Editor

TOAD’s Procedure Editor will display a hierarchy of all programming units within the source, as shown on the left-hand side of Figure 12. When a subroutine is selected, TOAD will immediately navigate to the selected subroutine. One drawback for any modem users: it seems as though TOAD recreates this list each time the user activates a file. One of our test environments accessed a database over a modem line. In this environment, we found that there could be wait times of 10 seconds simply to switch between multiple files that had already been opened. However, in normal LAN configurations, this should not be a concern.

TOAD's help documentation is very easy to follow. It uses the standard Windows help utility. It has the standard table of contents, index, and Find capabilities. We were very pleased with the tool help system. Help with PL/SQL syntax is available with the optional Knowledge Base (\$390). As mentioned in the SQL Navigator review, this is very comprehensive as well.

TOAD supports an internal source control system where backup copies are apparently saved to the database. It also supports SourceSafe versions 5.0 and later. We made a brief attempt at understanding the internal source control system, without success. However, given that our office uses VSS, we quickly moved on to setup of that environment. We had a few minor difficulties with the SourceSafe integration, but nothing major. TOAD supports VSS Check-out, Check-in, and Undo Check-out, utilizing a graphical interface very similar to the stand-alone VSS client. You can navigate the source trees and view all files and their checked-out status.

TOAD has extensive tuning options. It provides base functionality, such as display of Explain Plans and server statistics. For users who wish to perform more extensive tuning, Quest sells an optional add-on, the Xpert Tuning module. From our brief exposure, it appears to very easily allow the user to view a SQL command using different hint options and the statistics for each version of the command.

TOAD also has debugging capabilities available with the optional PL/SQL Debugger. The debugger seems to have a different set of features than most developing tools. For instance, we did not find the standard "Step Into" feature. However, it does list the capability to set breakpoints on loop iterations, which many products do not have. Unfortunately we were unable to use the debugger features in our trial version to verify these features.

As with most of the products, TOAD includes color coding of PL/SQL syntax. The color coding is user-configurable with a full color palette available. In fact there are many customization possibilities within this tool that cover nearly every aspect of how you would like the program to behave. However, like many of the other products we reviewed, TOAD does not provide an open interface to third party developers.

## **SQL PROGRAMMER**

We downloaded a trial version of SQL-Programmer, distributed by Sylvain Faust International. This product can be used with Oracle, Microsoft SQL-Server, and/or Sybase. According to [www.sfi-software.com](http://www.sfi-software.com), a single-user license of SQL-Programmer 2001 for Oracle is \$400 (Entry Level version) or \$1245 for the Expert Edition.

System requirements are Windows NT, 98, or higher and 30 megabytes of free disk space. You must have installed Internet Explorer 5.0 or higher and 32-bit SQL\*Net.

Installation was fairly easy. We requested a trial copy of SQL-Programmer at SFI's website. We then received an email detailing a download location and a trial license key. We downloaded and executed the setup program and entered the trial key information.

SQL-Programmer has some very nice features, but in our opinion, they are overshadowed by the non-intuitive interface. The lack of adequate help documentation compounds this problem.

Like TOAD, SQL-Programmer will not permit you to open a file unless you are already connected to a database. This can be very annoying, especially if a developer wants to edit off-line, such as on a laptop. Another annoying feature is that several toolbar buttons cannot be clicked on directly. Instead, one has to click on a small arrow to the right of the toolbar button.

Unlike most of the products we evaluated, SQL-Programmer does not provide a way to navigate directly to a specific procedure in a package. This means the developer is left with the old standby, the standard "Find" function. While this method works, it is inefficient. Also we find it helpful to have a list of the subroutines in our current document at easy reach. SQL-Programmer does show such a list for database packages, but not for text files currently under development.

The help documentation is extensive, but not actually very helpful. For instance, we found many help pages describing several different types of SPDWs. However, it took a lot more searching before we found a page describing what "SPDW" actually stood for (FYI, it's "SQL Programmer Development Window"). When we attempted to install version control management, the help documentation kept referencing an option in the "Options dialog window", but did not state where this dialog was located. We also accessed the online help at <http://www.sfi-software.com>. The online FAQ addresses some basic questions, but not as many as we would have liked. When we tried using the web site's search feature, we found a lot of broken links, primarily with the "Manuals" section.

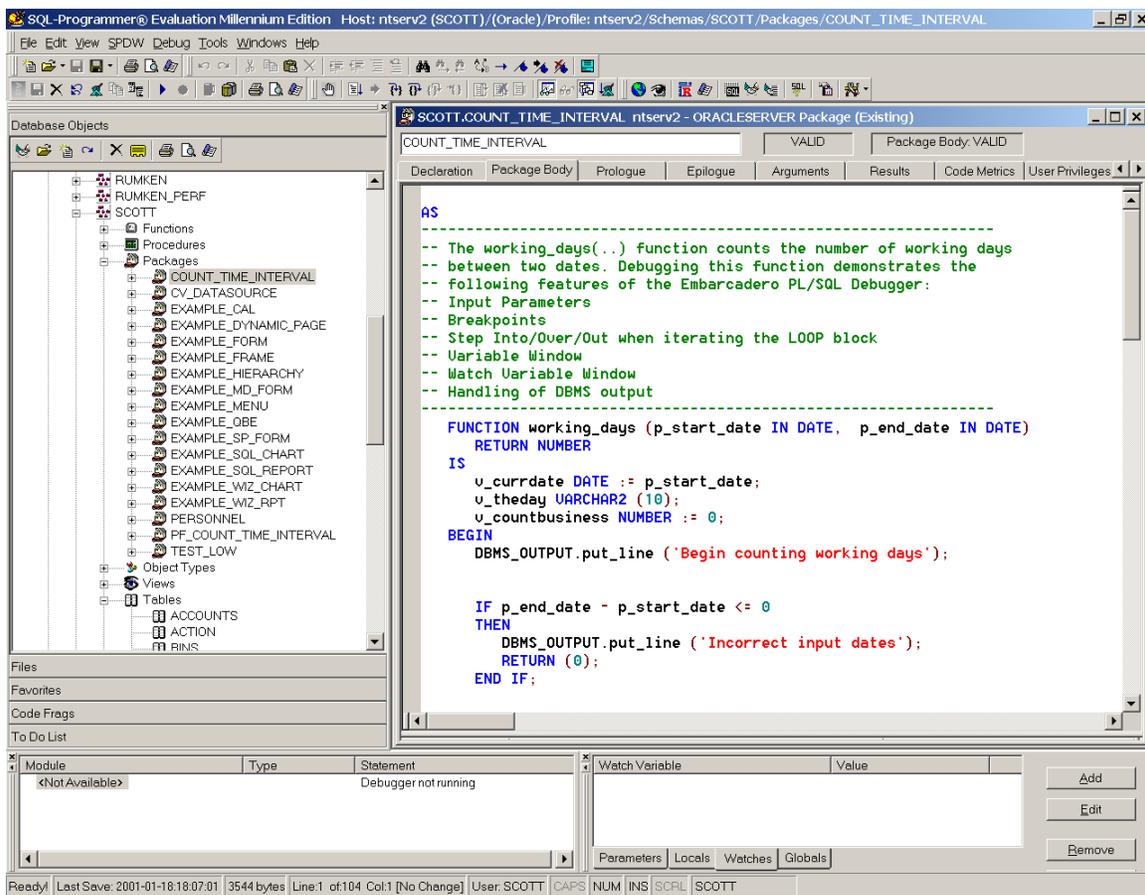


Figure 13 – SQL Programmer Debugger

We are also concerned with the stability of the product. SQL-Programmer does not handle a lost database connection well. Quite a number of error messages pop up, and all open files will close. In at least one case, our unsaved changes in an open text file were lost. Several times, the entire product shut down when the connection was broken. We also received frequent error messages during normal execution. These messages were cryptic and more reminiscent of debug messages. As end users, these error messages did little to explain how to resolve the problems.

SQL-Programmer's documentation states that the product integrates with SourceSafe version 6.0, PVCS version 6.2, StarBase's StarTeam version 4.1, and MKS Source Integrity. However, we were unable to determine how to activate the source control management due to the inadequate help documentation. We finally located the "Options" dialog that the help mentioned, but when we tried to activate the source control, we could not proceed past an error message that seemed to indicate that we were to change a non-editable field. After having struggled for so long attempting to enable this functionality, we finally gave up. If you wish to use SQL-Programmer's source control management, we suggest contacting SFI for support.

SQL-Programmer allows viewing of data, although it takes more steps than we'd like. We did not see an easy way to enable editing of a table's data. The interface did provide an easy way to modify a table's structure, however. This was pleasantly refreshing.

SQL-Programmer's tuning capabilities with their Oracle product seem pretty limited. There is a "Show Plan" option. However, we were not able to get this feature to work. The help hinted that this may be an option designed for Microsoft's SQL Server.

SQL-Programmer's debugging capabilities are standard (Figure 13). The utility allows stepping through code, watching and setting of variables, and setting breakpoints. However, we had difficulties with the debugger, as on two separate occasions the debugger simply stopped working. The first time, it would no longer permit us to compile for debugging. We eventually closed the SPDW and reopened the object for editing, at which time the debugger capabilities were available again. The second time, it simply stopped providing the watch variable values.

Like most of the products we reviewed, SQL-Programmer includes color coding of PL/SQL syntax. The color coding is user-configurable, but the developer is limited to a 16-color palette.

On the plus side, SQL-Programmer has a drag and drop style interface to help developers create select statements and call database subroutines. This interface is very easy and intuitive to use. We didn't see a method for creating update and delete statements. There is also a "Help SQL" feature that will display the syntax for built-ins and various PL/SQL structures, such as FOR-LOOPS. There does not seem to be a way to drop that syntax into the editor, however. So you'll need to use copy and paste for the syntax of these non-database objects.

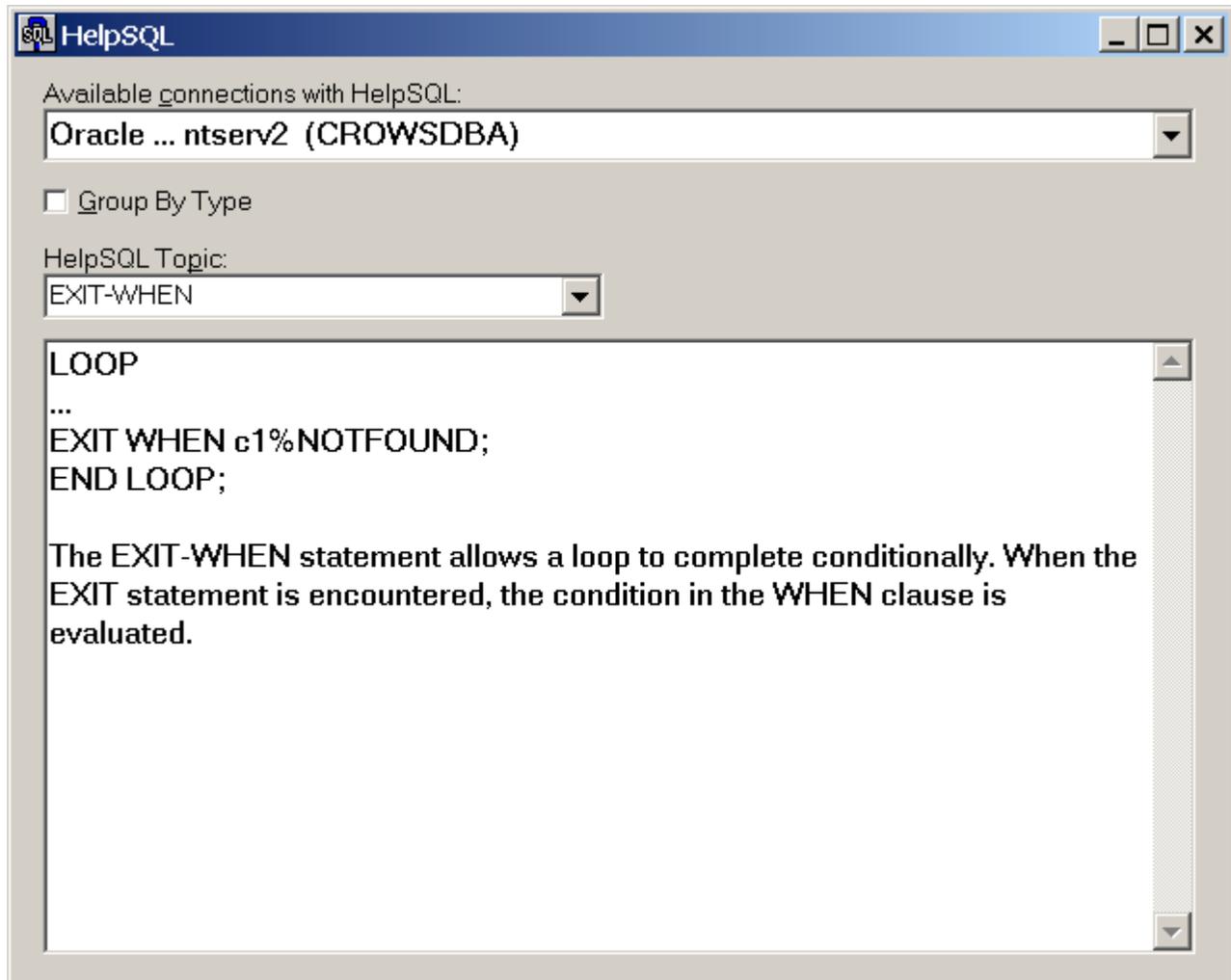


Figure 14 – SQL Programmer HelpSQL

SQL-Programmer has a SQL Scriptor tool that will create DDL scripts for your databases. According to the documentation I received from SFI, the Expert Edition is capable of creating scripts for your entire database with just a button-click. When we tested it ourselves, we found that it can take a very long time to run as you would expect when you reverse engineer an Oracle schema. So if you have need of reverse-engineering an entire schema or database, this is a product worth investigating further if you do not already have another tool with this functionality.

And finally, SQL-Programmer has a rare feature: a Virtual Editor. This feature allows developers to make and test changes to database subroutines without affecting other users. You can compile your subroutine in the Virtual Editor, test it, and view the end value of any output parameters. In the meantime, all other users would continue to see and use the original version until you choose to save your changes to the database. This is akin to the manner in which data changes are not visible to other users until the original user commits the data. We think this feature has potential to be very useful, especially in

environments with a limited number of test database instances. We would discourage its use in a production environment, but feel it has tremendous potential for development environments with many developers. Unfortunately, this feature seems to have a formidable drawback. The debugging features do not seem to be enabled with the Virtual Editor. Therefore, you cannot use the Virtual Editor where it would probably be most useful – to debug an errant value in a newly changed routine while leaving in place the fully-tested version for other users. We attempted to obtain more information on the Virtual Editor, to determine if this was an inherent limitation. Unfortunately, when we followed the link within the help documentation for more information on this feature, no more information was to be found. We tried executing a help search for the phrase “Virtual Editor”, but the only hit was the original page we had already seen. We think this feature has great potential, but we’d like to see SFI enable the debugging functionality with it.

## **EVALUATION METRICS**

Each of the products was evaluated based on a comparison of the following metrics.

- Cost – List price for a single seat user license. We did not investigate discount possibilities which some vendors may provide.
- GUI Environment – Graphical user interface support which allows the user to navigate through the tool with a mouse.
- Object Browser – Browse Oracle objects with a MS Windows explorer type tool.
- Code Templates – An interface that supplies the code syntax for development. Examples are: a cursor for loop, a select statement, and an Oracle function.
- Code Formatting – Formats your PL/SQL code to provide consistent indentation format, consistent case of reserved words, etc.
- Code Assistant – The automatic prompting of parameters, or package interface objects like variables, procedures and constant.
- Version Control – Integration with a source code version management system.
- Tool Help – Help with the tool windows and dialogs. A measure of how well the help system is integrated into the tool. Also measures the quality of the user manual.
- PL/SQL Help – Help with the underlying PL/SQL syntax and usage.
- Plan Analysis – Provides an easy to use interface to the Oracle explain plan functionality.
- Test / Debugging – An indication of how easy it is to debug PL/SQL code. Features that better the products score would include a debugger with support for single step, step into, break on line, break on condition, or break on iteration, the dynamic setting and watching of variable values etc.
- SQL Support – Features here would include being able to view data, views, table properties, etc. from within the development tool. Also helpful here would be features that would allow software users edit access to non PL/SQL objects like tables and views.
- Ease of Use – How easy the package is to use. A good score here would suggest an intuitive, flexible interface in which the user may never need to refer to the product help and user documentation. Also helpful to this score is an environment in which personal preferences can be easily customized. In our view, this is the most important metric.
- Third Party Support – An interface that third party developers can use to extend and or enhance the product reviewed. The plug-in feature within the Netscape and the Microsoft Internet Explorer browsers are examples of this type of flexibility.

## **PRODUCT COMPARISON MATRIX**

We took the individual evaluation metrics and created the following comparison matrix. This matrix includes all the reviewed products except for Expediter/SQL which we disqualified from this review.

Product	SQL*PLUS and a text editor	Oracle Procedure Builder	SQL Programmer	TOAD	RapidSQL	SQL Navigator	PL/SQL Developer
Cost	Shipped with Server	\$4,995 (1)	\$1245 Expert Edition	\$670 base tool + \$520 Debugger	\$695 base tool + \$395 Debugger	\$670 base tool + \$520 Debugger	\$150
GUI Environment	No	Yes	Yes	Yes	Yes	Yes	Yes
Object Browser	No	Yes	Yes	Yes	Yes	Yes	Yes
Code Templates	No	Yes	Limited	Yes	Yes	Yes	Yes
Code Formatting	No	No	Yes	Yes (2)	No	Yes (2)	Yes (2)
Code Assistant	No	No	No	No	No	No	Yes
Version Control	No	Yes (3)	Yes (4)	Yes	Yes	Yes	Yes
Tool Help	Limited	Full	Full (5)	Full	Full	Full	Word Document
PL/SQL Help	No	Yes	Yes	Yes	Limited	Yes	Yes
Plan Analysis	No	No	Yes (4)	Yes (6)	Yes	Yes	Yes
Test / Debugging	No	Yes	Yes	Yes	Yes	Yes	Yes
SQL Support	Manual	Triggers and PL/SQL	Full (7)	Full	Full (7)	Full	Full
Server Install Required	No	No	Yes	Yes	No (8)	Yes	No
Ease of Use	1	3	3	3	4	4	5
Third Party Tool Support	No	No	No	Limited to "partners"	No	Limited to "partners"	Yes

Table 2. Product Comparison Matrix

Notes:

1. Part of the Internet Developer Suite.
2. With Reveal Net Option priced between \$125 and \$150.
3. PVCS and Clearcase only
4. Documented feature which we were not able to get to work.
5. Full help documentation exists, but we feel it isn't written well.
6. Internal and SourceSafe only
7. Does not support editing data in a table.
8. If you install the profiler, then server side code needs to be installed.

## **CONCLUSION**

All of this analysis leaves us with the question we started with, which product will "supercharge" our development efforts. So here's the scoop. Our favorite product is Allround Automation's PL/SQL Developer. It has a number of good things going for it. First, except for SQL\*Plus, it is the cheapest of the products we reviewed. We liked that a lot. Second, it is focused on PL/SQL development. This focus gives it a very clear and easy to understand interface that is dedicated to solving your

PL/SQL development needs. This focus is best illustrated with the tool manual, which describes all of the tool features, is excellently written, and is easy to understand. Some of the other tools we reviewed had interfaces that were very busy, which in turn required us to spend a lot of time reading a lot of detailed help system pages before we were able to freely move around in the environment.

Our runner up products, SQL Navigator and RapidSQL, will also supercharge your PL/SQL development. We didn't select either as our favorite product, as each is considerably more expensive and, while both are nice products, each is slightly more difficult to use than PL/SQL Developer. Given a different set of requirements, such as having to support more than just Oracle databases, having to support Java and Web development, or having to focus extensively on tuning issues, we could easily have chosen one of these tools over our PL/SQL Developer selection.

Our final, and perhaps most important, piece of advice to you is that if you are currently using SQL\*Plus to create and maintain your PL/SQL code, check out the products in this review as they all offer productivity improvements over our reference SQL\*Plus environment.