

ENABLING AGILE DATABASE DEVELOPMENT WITH TOAD[®]

**Eliminate the bottleneck in your
DevOps workflow with Toad for
Oracle Developer Edition and
Toad DevOps Toolkit**

By John Pocknell, Senior Product Manager, Quest[®]

Quest[®]



Table of Contents

Introduction	3
Part 1: Improve your functional correctness, code quality, code maintainability and application performance	4
Part 2: Implementing and configuring Toad Developer Edition for Oracle database developers	11
Part 3: Integrating Toad DevOps Toolkit with your continuous integration servers	18
Conclusion	25



Introduction

What does it take to make your database development as agile as your application development?

The advantages of agile development include shorter time to value, lower risk and greater flexibility. In fact, most application developers already consider agile a mainstream approach. But because databases and applications are different animals, developers have been slower to embrace agile, especially in relational environments.

For instance, if application developers deploy version 2 of an app and there's a problem with it, they can easily restore version 1 so users can continue working while version 2 gets fixed. A database, on the other hand, is more like a living organism, with a current state that must always be managed to maintain the integrity of the data. Simply overwriting version 1 of the database schema with version 2 could easily result in data loss.

Database changes are usually made carefully, methodically and with slow, manual processes like scripting to avoid the risk of costly system downtime. However, clinging to traditional database management practices comes at a very high price. Since people don't scale in a linear fashion, those manual processes can never scale to the level required to support agile projects. The processes become a bottleneck that keeps many organizations from reaping the full benefits of agile development.

This e-book includes walk-throughs, implementation guidelines and links to videos that will show you how to use [Toad® for Oracle Developer Edition](#) and [Toad DevOps Toolkit](#) to automate your database development processes and realize the full promise of agile: the ability to release software in prompt response to market changes.

PART 1

Improve your functional correctness, code quality, code maintainability and application performance





Before most database developers are ready to come into the agile fold, they need to be convinced that automation tools can enable them to shorten release cycles while still minimizing risk in production. The right tools will support agile development while also supporting four important areas of Oracle database development:

- **Functional correctness** — Ensuring all code is tested against all application use cases to reduce the risk of costly defects
- **Code quality** — Consistently applying best-practice coding standards to minimize the likelihood of unplanned development cycles
- **Code maintainability** — Enabling developers to better understand code constructs and reduce the risk of time-consuming code changes and errors
- **Application performance** — Making code run faster to reduce the risk of slow response time and business interruptions and to meet required service level agreements (SLAs)

Part 1 of this e-book explains how Toad for Oracle Developer Edition supports those areas. It examines them across six steps in the typical database development process:

1. Accessing the source code repository
2. Creating and maintaining code
3. Testing code
4. Code review
5. Optimizing code performance
6. Integration testing

Accessing the source code repository

In order to establish and maintain code integrity, most development shops store PL/SQL code and other objects as files in a centralized code repository using a version control system (VCS). Toad's Team Coding utility streamlines the versioning process by enabling developers to check code out and back in directly from the Toad Editor simply by accessing the objects in the database, without the need to access the files stored in the VCS. Team Coding ensures the database objects and the corresponding files in the VCS are in sync. If any differences between the database and VCS versions are detected at the point of checkout, Team Coding will display a warning message so the developer can investigate the issue or merge the differences into a new version.

Creating and maintaining code

The Toad Editor is the perfect tool for creating and maintaining SQL, SQL*Plus and PL/SQL code. It includes many time-saving productivity devices, including an advanced PL/SQL formatter, code snippets, SQL recall and code templates. Code templates can be shared with other team members.



Testing code

DEFINING TEST CASES

If all you have are manual processes, PL/SQL unit testing is often woefully inadequate. Because it's so much work to write test cases, keep them in sync with the application as it changes, verify the results of the tests and so on, testing gets short shrift. And the frequent releases in an agile environment only exacerbate the problem.

The Code Tester for Oracle component of the Toad for Oracle Developer Edition facilitates the adoption of agile methodologies by streamlining the entire testing process. With Code Tester, you don't write test code; you simply describe your tests based on your use cases, and the tool then generates the test code as a PL/SQL package that implements your test definition (see Figure 1). All tests are stored in the Code Tester repository so they can be reused as part of a regression cycle. Not only do developers save hours of time writing the test code, but the design of the code unit more closely matches the functional requirements of the application, since the developers have to think carefully about how it will be used.

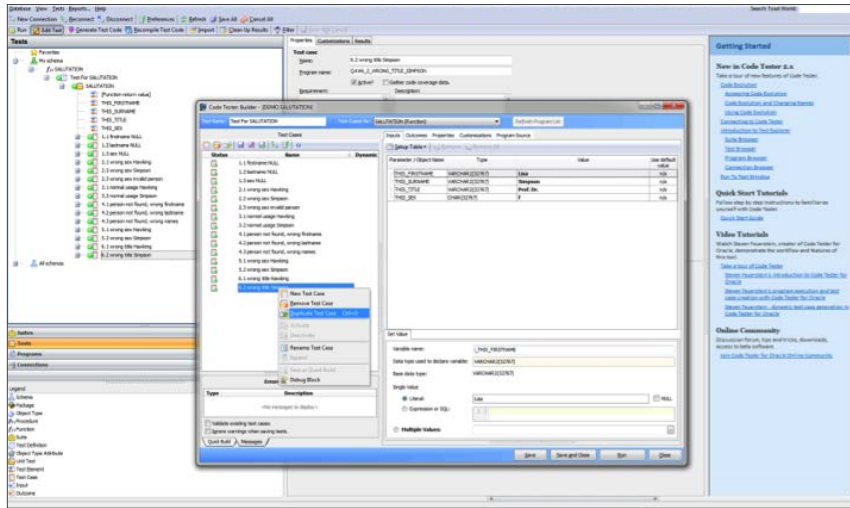


Figure 1. Code Tester for Oracle automatically generates test code based on your use cases.

The Code Tester for Oracle component of the Toad for Oracle Developer Edition facilitates the adoption of agile methodologies by streamlining the entire PL/SQL unit testing process.

In addition, if you want to require developers to regression test their PL/SQL code before they check it into version control, you can do so in the Team Coding settings.

Code review

Code reviews ensure consistent adherence to best practices and quality standards in order to reduce problems in production and improve the maintainability of the code for the future. Toad's Code Analysis streamlines code reviews by analyzing PL/SQL code against a library of rules designed to ensure high code quality. Toad includes nearly 200 predefined rules, which you can modify to meet your needs. Each team can select a subset of the rules and store them in a rule set for their project.

Developers can see and correct their code violations right in the Editor, as shown in Figure 2, or they can view a summary of coding violations across a schema or project.

In addition, if you want to require developers to review any PL/SQL code that fails to meet minimum standards before they check it into version control, you can do so in the Team Coding settings.

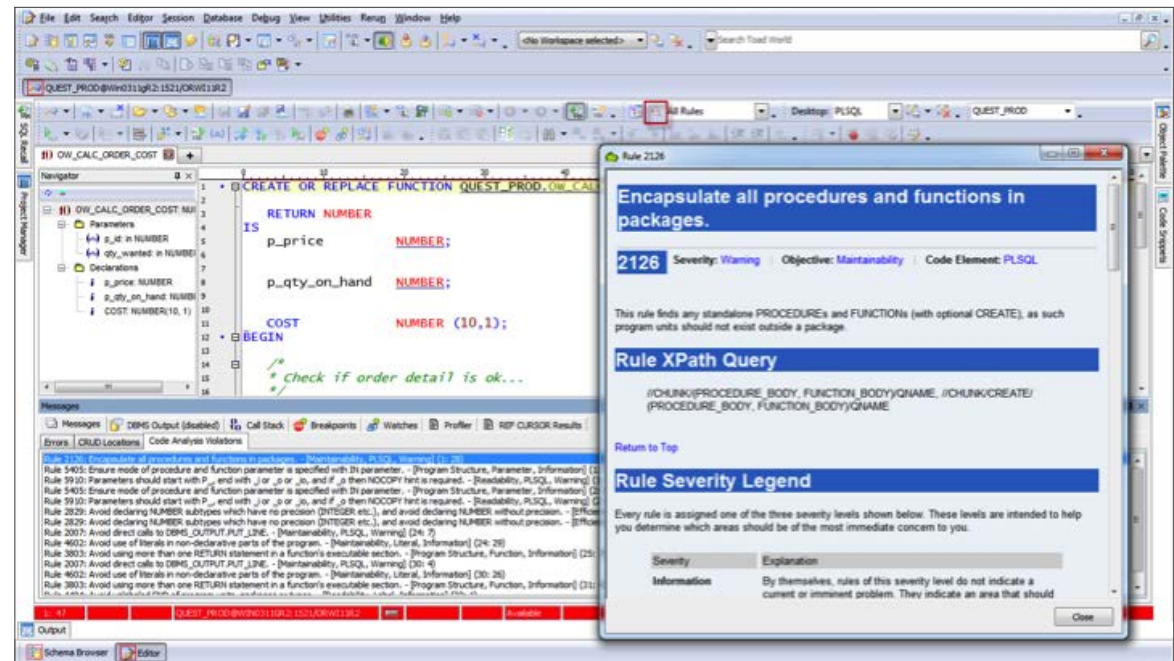


Figure 2. Database developers can see and correct coding violations right in the Toad Editor.

Optimizing code performance

Trying to optimize code performance manually is a challenging and time-consuming exercise, ill-suited for the rapid development cycles in an agile environment. SQL Optimizer for Oracle offers a fully automated approach to maximizing SQL and PL/SQL code performance — again, right from the Toad Editor. Simply select a SQL statement (even one inside a PL/SQL stored procedure), click the Auto Optimize SQL button, and Toad will begin looking for rewrites of the original SQL statement, automatically eliminating those with identical execution plans (see Figure 3). Comparison windows enable you to compare graphical performance statistics, execution plans and SQL syntax for each alternative in order to make an informed choice about the best statement to use.

SQL Optimizer will also proactively identify which SQL statements could slow your application. Its Scan SQL component scans database objects and source code files to identify potentially problematic SQL statements, and then classifies them by severity so you can focus your tuning efforts with SQL Optimizer more effectively.

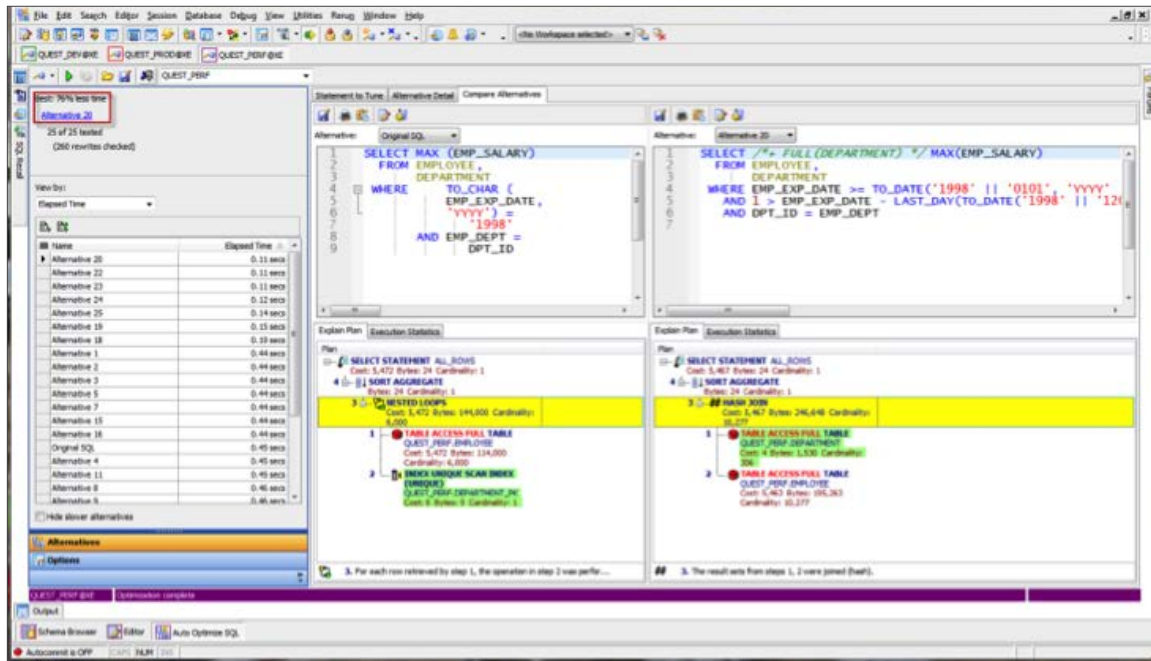


Figure 3. Toad's Auto Optimize SQL is the most effective way for developers to tune PL/SQL and SQL.

Integration testing

PL/SQL REGRESSION TESTING

Agile development can involve frequent updates to code, so automating the process of regression testing is critical. With Toad, because unit tests are stored in a repository, you can easily run Code Tester over all the code changes in a sprint to see whether code defects were introduced.

CODE PERFORMANCE TESTING

Additional development cycles often occur post-production because SQL statements and PL/SQL code did not scale as expected in production. The Benchmark Factory for Oracle component of the Toad for Oracle Developer Edition helps you avoid this problem by analyzing whether a given SQL statement or PL/SQL code unit is likely to meet your performance or SLA expectations in production. For example, the tool could produce a graph showing how increasing the user load affects response time.



PART 2

Implementing and configuring Toad Developer Edition for Oracle database developers

The road to agile database development leads away from traditional, manual processes and toward fully automated pipelines that streamline work and minimize the risk of data loss or downtime.

Part 2 of this e-book focuses on implementing and configuring the various components of Toad for Oracle Developer Edition and covers some of the key developer tasks such as:

- Access to source control
- PL/SQL unit testing
- Code reviews
- SQL and PL/SQL optimization



Setting up the environment

As depicted in Figure 4, Toad for Oracle Developer Edition will be installed on staff desktops. It also requires some Toad repositories to be installed on your database and integration with version control systems.

Toad DevOps Toolkit will be installed adjacent to your build automation server (such as Jenkins or Bamboo. See Part 3 of this e-book for more details on setting up Toad DevOps Toolkit.

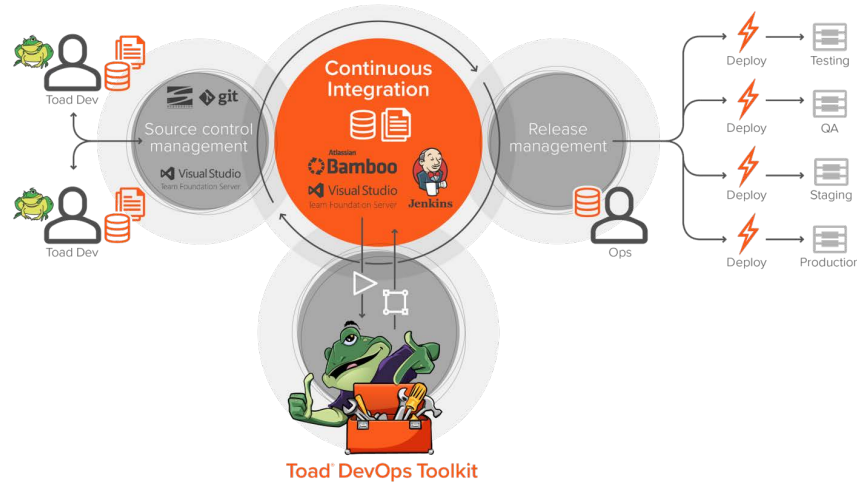


Figure 4 How Toad for Oracle Developer Edition and Toad DevOps Toolkit integrate with a typical DevOps infrastructure.

Installing the developer desktop

First, you install Toad for Oracle Developer Edition on your developers' Windows desktop or laptop machines. It includes the following products:

- Toad for Oracle Xpert Edition (includes SQL Optimizer for Oracle)
- Code Tester for Oracle
- Benchmark Factory for Oracle
- Toad Data Modeler

Toad for Oracle Developer Edition can also work with Oracle Instant Client.

The implementation also requires an Oracle client for Windows. The client should match the Oracle server in version number and bit-width (32 vs. 64) as nearly as possible. Toad for Oracle Developer Edition can also work with Oracle Instant Client.

Toad for Oracle itself can be installed directly on the Windows desktop using the web installer or a network package with silent installation.

IMPLEMENTING TEAM CODING

For improved collaboration, Team Coding is a utility that integrates with your version control system. Developers access PL/SQL code and other objects from the Oracle database as usual, but Team Coding controls and monitors the check-out/check-in of the corresponding files in source control. Team Coding supports the following VCS providers:

- Serena PVCS Version Manager
- Microsoft Visual SourceSafe
- Microsoft Team Foundation Server
- Mercurial (from Toad 12.10)
- IBM Rational ClearCase
- Perforce
- CVS
- Git
- Subversion

Team Coding requires a set of repository tables to be installed in a schema in the Oracle database in order for the required selection of schema objects and their corresponding files in version control to be properly managed and synchronized.

For more information on setting up Team Coding with VCS integration, [watch this video.](#)

For information on the usage of Team Coding with VCS, [watch this video.](#)

IMPLEMENTING CODE ANALYSIS

Code Analysis is a rules-based feature for code review. It comes with approximately 200 pre-defined coding rules across a number of standard category Rule Sets such as Program Structure, Maintainability and Efficiency. As shown in Figure 5, you can use your own coding standards and create your own Rule Sets, then share the rules across the members of each team. Code Analysis also includes a repository for storing the results of your code reviews. For more information on setting up and using Code Analysis, [watch this video.](#)

You can select an option in Team Coding that will require a code review whenever developers try to check their code into source control. Toad for Oracle will prevent any code that does not meet the required quality levels from being checked in until the problems are fixed. For more information on setting Team Coding policies, [watch this video.](#)

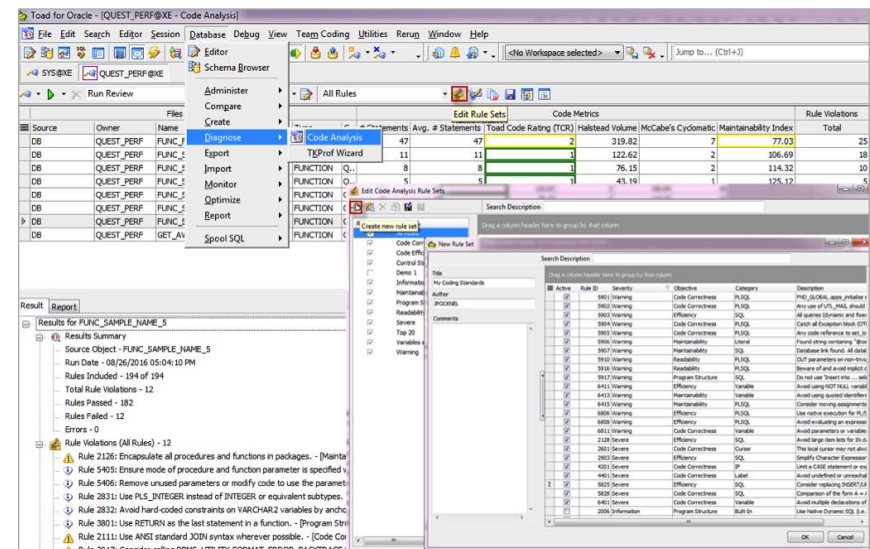


Figure 5: Creating a new Rule Set in Code Analysis and selecting rules from the library.

OPTIMIZING SQL AND PL/SQL

When a PL/SQL program runs slowly, you need to be able to pinpoint the bottleneck. If the problem is a poorly written SQL statement, you then need to improve its execution performance. Toad and SQL Optimizer handle the process in two steps.

STEP 1 – PROFILING PL/SQL

The PL/SQL Profiler feature in Toad works in conjunction with the Editor to time the execution of each line of PL/SQL code as the program runs. PL/SQL Profiler stores the data in a repository and represents it graphically, as shown in Figure 6. Any performance bottlenecks become immediately apparent. For more information on setting up and using PL/SQL Profiler, [watch this video](#).

STEP 2 – OPTIMIZING THE SQL

Next, use SQL Optimizer in Toad to relieve the bottlenecks. SQL Optimizer automatically reforms the statement to remove any PL/SQL-specific elements, such as INTO statements, and converts locally declared variables to bind variables.

The Auto Optimize SQL feature (see Figure 7) will then start generating multiple rewrites of your original SQL statement until it finds one that executes more quickly. It presents execution plans and statistics to help you select the best alternate statement. For more information on using Auto Optimize SQL, [watch this video](#).

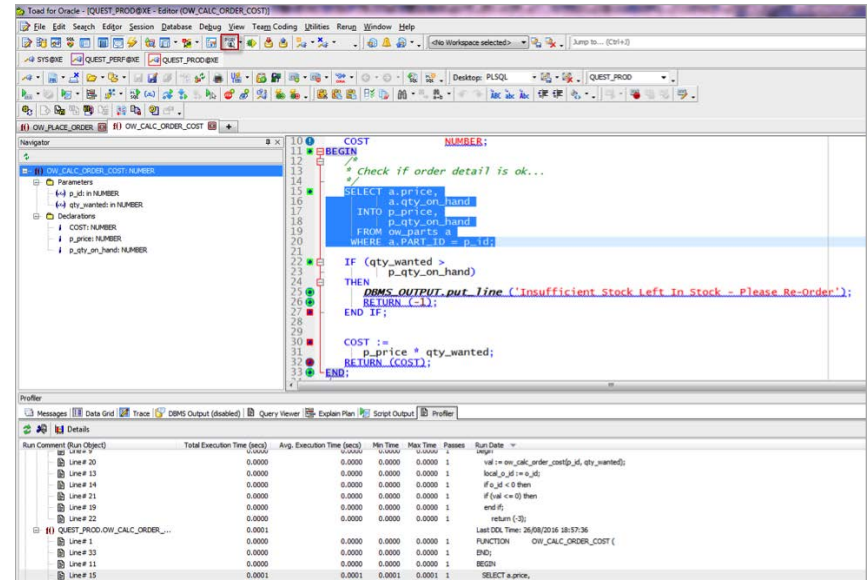


Figure 6: PL/SQL Profiler identifying a slow-running SQL statement inside a PL/SQL program.

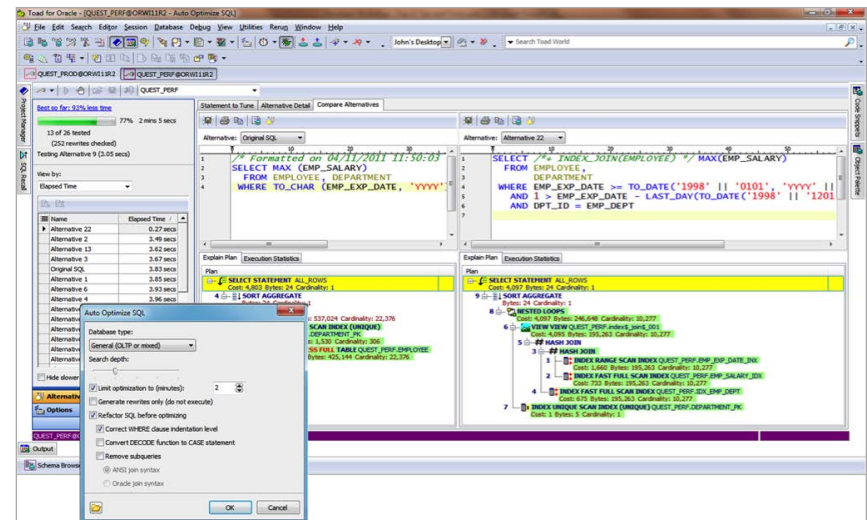


Figure 7: Auto Optimize SQL automatically finding a faster SQL alternative.

Implementing Code Tester for Oracle

Testing code is a double-edged sword. Developers know that testing is important, but it often involves writing lots of test code, keeping that test code in sync with changes in application code, verifying test results and taking other steps that effectively keep developers from their highest-value work.

Code Tester for Oracle makes it easy to define tests, generate test code and run tests, all within an easy-to-use graphical interface. Best of all, with Code Tester you don't write test code. You describe your tests based on your use cases, and the tool generates test code as a PL/SQL package that implements your test definition.

Code Tester requires the installation of a repository to store all your test definitions and test cases. This repository will be accessed whether you execute the tests directly from the Toad Editor or from Code Tester itself.

The Code Tester repository needs to be installed on the same database where your developers will be defining their tests. You can enable public or private access to the repository depending on your requirements.

With Code Tester, you don't write test code. You describe your tests and the tool generates test code that implements your test definition.

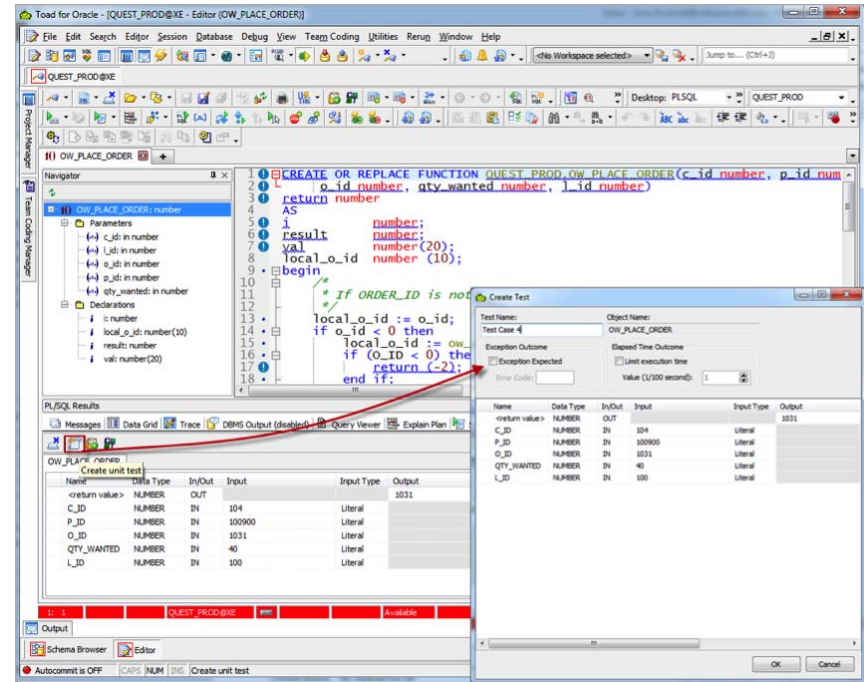


Figure 8: Creating a PL/SQL unit test in Toad

CREATING UNIT TESTS IN TOAD EDITOR

As shown in Figure 8, you can create unit tests directly in Toad for Oracle and store them in the Code Tester repository. The process creates reusable tests that will be stored for ongoing regression testing throughout the lifetime of the code.

In Toad Editor, when you execute the code normally, Code Tester uses any input and output values to create the unit test definition, which it then stores in its repository. For more information on creating PL/SQL unit tests in Toad, [watch this video](#).

CREATING UNIT TESTS IN CODE TESTER

In the Code Tester UI itself, the simplest way to create unit tests is to use the Run-to-Test feature, as shown in Figure 9. It provides more comprehensive options for defining input values and expected outcomes.

As noted above, all unit tests, whether created in Toad or Code Tester, will be stored in the Code Tester repository. There you can view all your tests, test suites and PL/SQL code (see Figure 10).

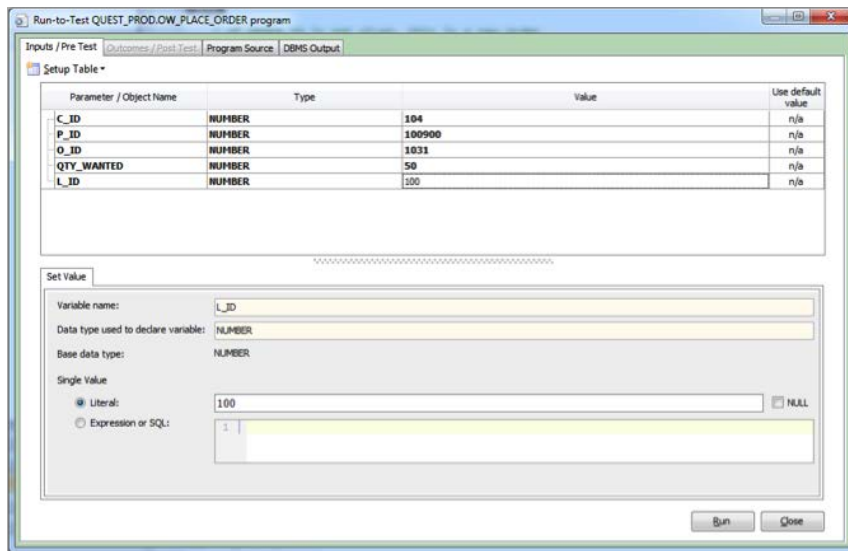


Figure 9: Creating a PL/SQL unit test in Code Tester using Run-to-Test

For more information on creating comprehensive PL/SQL unit tests in Code Tester, [watch this video](#).

SHARING AND CONTINUOUS INTEGRATION

The next step after automating these manual tasks is to make them available to all database developers and for continuous integration through a central repository. Part 3 of this e-book shows you how to set that up.

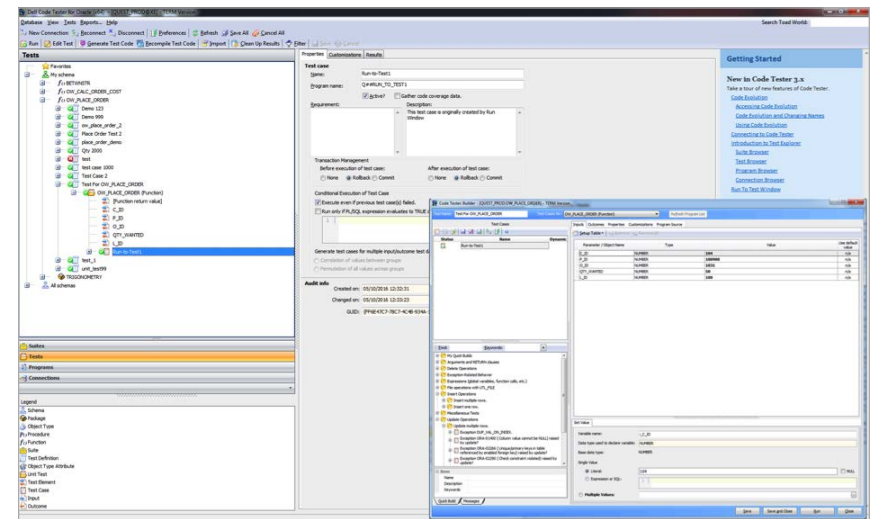


Figure 10: Code Tester main window showing Test Builder (inset)

PART 3

Integrating Toad DevOps Toolkit with your Continuous Integration servers



Collaboration and automation play a big role in agile database development. Once you've enjoyed the productivity boost from sharing and viewing queries, files, objects and data sets with other database developers in real time, you'll never go back.

Part 3 of this e-book is a setup guide for Toad DevOps Toolkit which will sit adjacent to your Continuous Integration server and enable you to call Oracle database development tasks that were setup in Toad for Oracle.

Toad DevOps Toolkit comprises six modules:

- Code Analysis
- Code Tester
- Database Compare and Sync
- Schema Compare and Sync
- Data Compare and Sync
- Script Execution

Calling code reviews on Toad DevOps Toolkit from a CI server.

Toad DevOps Toolkit effectively becomes an extension to your automated build process because it lets you include automated static code review and analysis performed on a CI server such as Jenkins, Bamboo or Team City.

Toad DevOps Toolkit will report build Pass/Fail notifications inside your build console so you know whether or not to proceed with the build. Code quality reports can also be generated and displayed inside your build console.

If you are using Jenkins, we provide a plug-in to the Jenkins build console which simplifies how you instruct Toad DevOps Toolkit to perform these tasks.

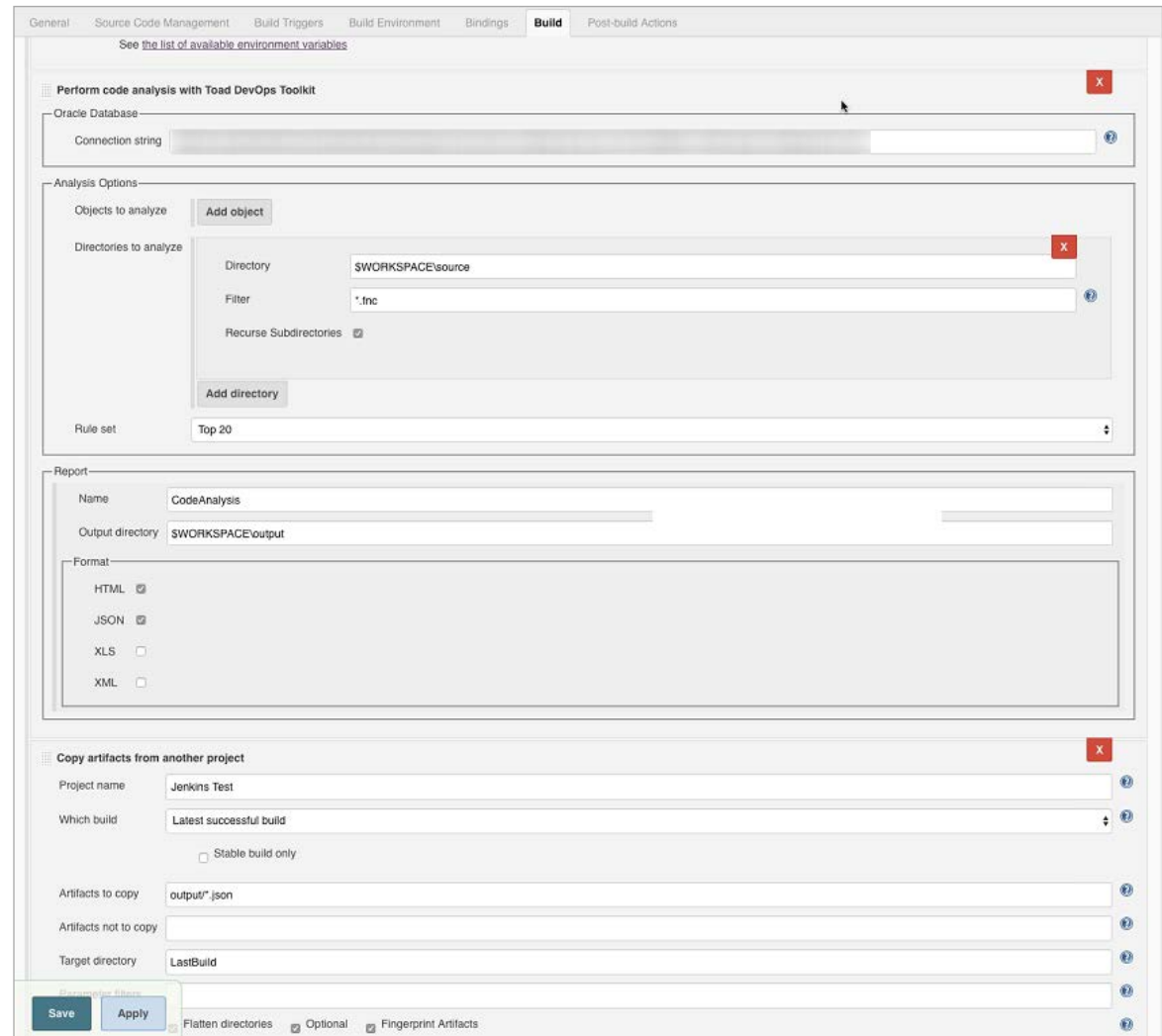


Figure 11 The Jenkins plug-in for Toad DevOps Toolkit simplifies the execution of database static code reviews.

Toad DevOps Toolkit extends your build automation process by including key Oracle database development and deployment tasks.

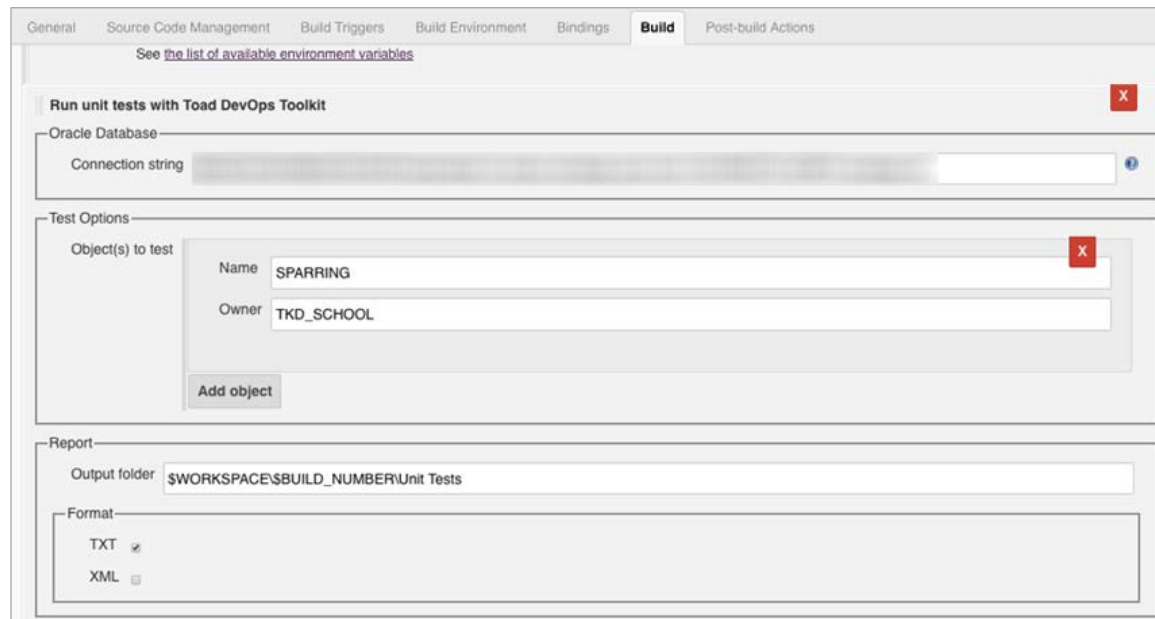


Figure 12 The Jenkins plug-in for Toad DevOps Toolkit simplifies the execution of PL/SQL unit tests.

Calling PL/SQL unit tests on Toad DevOps Toolkit from a CI server.

Toad DevOps Toolkit extends your automated build process because it lets you execute PL/SQL unit tests (final regression testing) against the code in the build on a CI server such as Jenkins, Bamboo or Team City.

Toad DevOps Toolkit will report build test Pass/Fail notifications inside your build console so you know whether or not to proceed with the build.

If you are using Jenkins, we provide a Toad DevOps Toolkit plug-in to the Jenkins build console which simplifies how you instruct Toad DevOps Toolkit to perform these tasks.



Calling a Database Compare on Toad DevOps Toolkit from a CI server.

Once you have a viable set of build artifacts, you are now ready to deploy your changes to your target database such as Test, QA or Staging.

Before you do that, it's important to understand whether the target Oracle database is configured the same as the Development (or source) database so that any testing that may be carried out, such as PL/SQL unit testing will exhibit predictable behavior.

From the CI server, you can initiate a Database Compare operation which will run inside Toad DevOps Toolkit and compare objects such as tablespaces, user roles, users, etc to see if they are the same. If not, then a synchronization DDL script will be generated which, when executed on the target database, will make the target the same as the source.

Calling a Schema Compare on Toad DevOps Toolkit from a CI server.

The next important aspect to understand is whether the target Oracle schema, into which you intend to deploy your database build artifacts, are the same as the Development (or source) database so that any data structure changes don't break any PL/SQL code program executions or other database transactions.

From the CI server, you can initiate a Schema Compare operation which will run inside Toad DevOps Toolkit and compare all the schema objects between source and target to see if they are the same. If not, then a synchronization DDL script will be generated which, when executed on the target database, will make the objects in the corresponding target schema the same as the source.

Calling a Table Data Compare on Toad DevOps Toolkit from a CI server.

Finally, you will need to understand whether all the table data in the target Oracle schema, into which you intend to deploy your changes, is the same as the Development (or source) database so that any data differences don't break any PL/SQL code program executions or other database transactions.

From the CI server, you can initiate a Data Compare operation which will run inside Toad DevOps Toolkit and compare the data in all the tables between source and target to see if they are the same. If not, then a synchronization DDL script will be generated which, when executed on the target database, will update all the table data in the corresponding target schema so it is the same as the source.



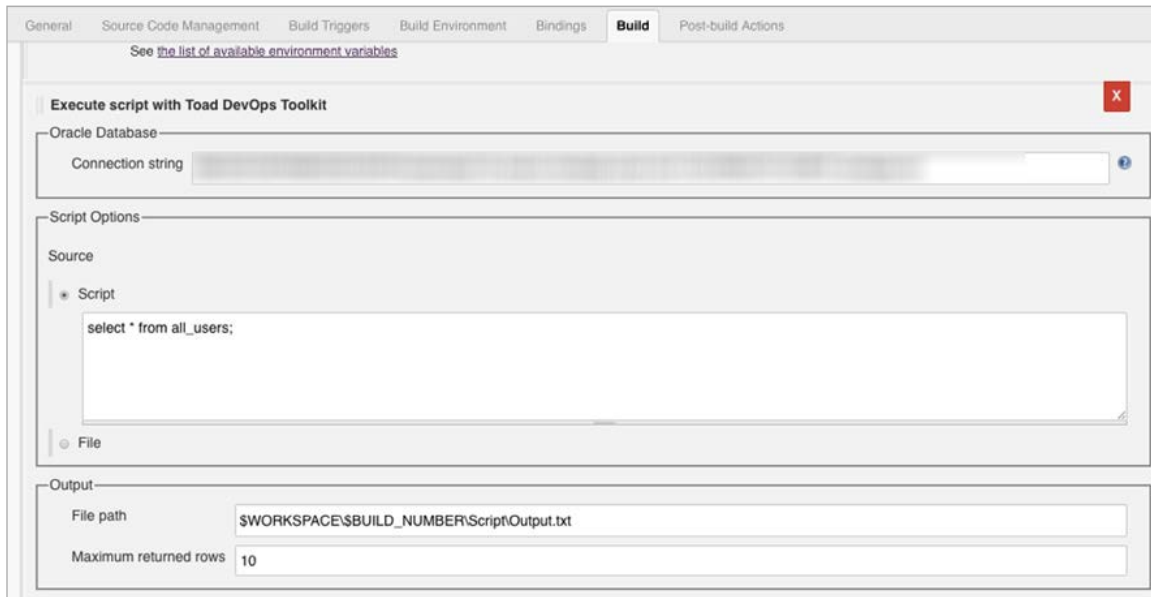


Figure 13 The Jenkins plug-in for Toad DevOps Toolkit simplifies the execution of your synchronization scripts.

The automatic execution of Oracle database tasks within Toad DevOps Toolkit is simplified through the use of a Jenkins plug-in to the Jenkins operations console. For other automation servers such as Bamboo and Team Foundation Server, a scripts library is available to perform the same tasks.

Calling a Script Execution on Toad DevOps Toolkit from a CI server.

If differences have been identified between the source database and the target, either with the configuration, the schema objects or the table data, Toad DevOps Toolkit will have generated one or more DDL synchronization scripts.

For the operations person, they can decide at what point they want to synchronize the two databases or they can have the sync scripts executed automatically. It's really up to the organization.

In any event, script execution is simple and can be initiated from inside the CI operations console, whether that be Jenkins, Bamboo or TFS.

If you are using Jenkins, the Jenkins plug-in for Toad DevOps Toolkit simplifies this task even further.



Conclusion

Agile requires a fundamental shift in the way database development teams work. Instead of relying on traditional, manual processes that become an intractable bottleneck, database developers must look for ways to create fully automated pipelines that streamline their work while also minimizing the risk of data loss or downtime.

Together, [Toad for Oracle Developer Edition](#) and [Toad DevOps Toolkit](#) deliver automated features like Code Analysis, Code Tester, Database Compare and Sync, Schema Compare and Sync, Data Compare and Sync, and Script Execution.

Database developers now have the opportunity to shorten development cycles, minimize risk and catch up with agile application development in their organization.

ABOUT THE AUTHOR

John Pocknell is a senior product manager at Quest Software. Based at the European headquarters in the U.K., John is responsible for the strategy and roadmap for the Toad portfolio of products worldwide. He has been with Quest Software since 2000, working in the database design, development and deployment product areas, and has run many Toad training courses for customers. John has spent over 16 years successfully evangelizing Toad to customers at various events around the world and he writes many blogs and papers on the Toad user community, [Toad World](#), as well as technical papers about Toad on the [Quest Software](#) website.

John has worked in IT for more than 30 years, most of that time in Oracle application design and development. He is a qualified aeronautical engineer with more than 10 years of experience in provisioning IT consultancy services and implementing quality assurance systems to ISO 9001.

ABOUT QUEST

At Quest, our purpose is to solve complex problems with simple solutions. We accomplish this with a philosophy focused on great products, great service and an overall goal of being simple to do business with. Our vision is to deliver technology that eliminates the need to choose between efficiency and effectiveness, which means you and your organization can spend less time on IT administration and more time on business innovation.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (www.quest.com) for regional and international office information.

© 2018 Quest Software Inc. ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal

Trademarks

Quest, Toad and the Quest logo are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit www.quest.com/legal/trademark-information.aspx. All other trademarks are property of their respective owners.