# ENABLING AGILE DATABASE DEVELOPMENT WITH TOAD®

## Eliminate the bottleneck in your agile process with Toad Development Suite for Oracle® and Toad Intelligence Central

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:** Implement the automation components of Toad Development Suite for Oracle **14**

**Part 3:** Set up Toad Intelligence Central to maintain coding standards and access continuous integration servers **21**

**Conclusion** **30**



Quest

# 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® Development Suite for Oracle and Toad Intelligence Central 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 will come into the agile fold, they have to see that automation tools enable them to shorten development cycles while still minimizing risk. 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 bugs and meet required service-level agreements (SLAs)
- **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

Part 1 of this e-book explains how Toad Development Suite for Oracle and Toad Intelligence Central support those areas. It examines them across eight steps in the typical database development process:

1. Preparing the development environment
2. Accessing the source code repository
3. Creating and maintaining code
4. Testing code
5. Debugging code
6. Code review
7. Optimizing code performance
8. Integration testing

# PREPARING THE DEVELOPMENT ENVIRONMENT

## ESTABLISHING A COHERENT, SHARED DEVELOPMENT ENVIRONMENT

The first step in the database development process is to establish a good environment. Toad Intelligence Central makes this simple task (see Figure 1). From the admin console, an administrator can create project folders and assign access rights according to groups and users defined manually or through Active Directory. Database developers can easily share relevant standards documents, code templates, scripts, automations and more. In fact, many windows in Toad Development Suite for Oracle can directly publish to and pull from Toad Intelligence Central, which helps ensure consistency, promote best practices and drive down errors.

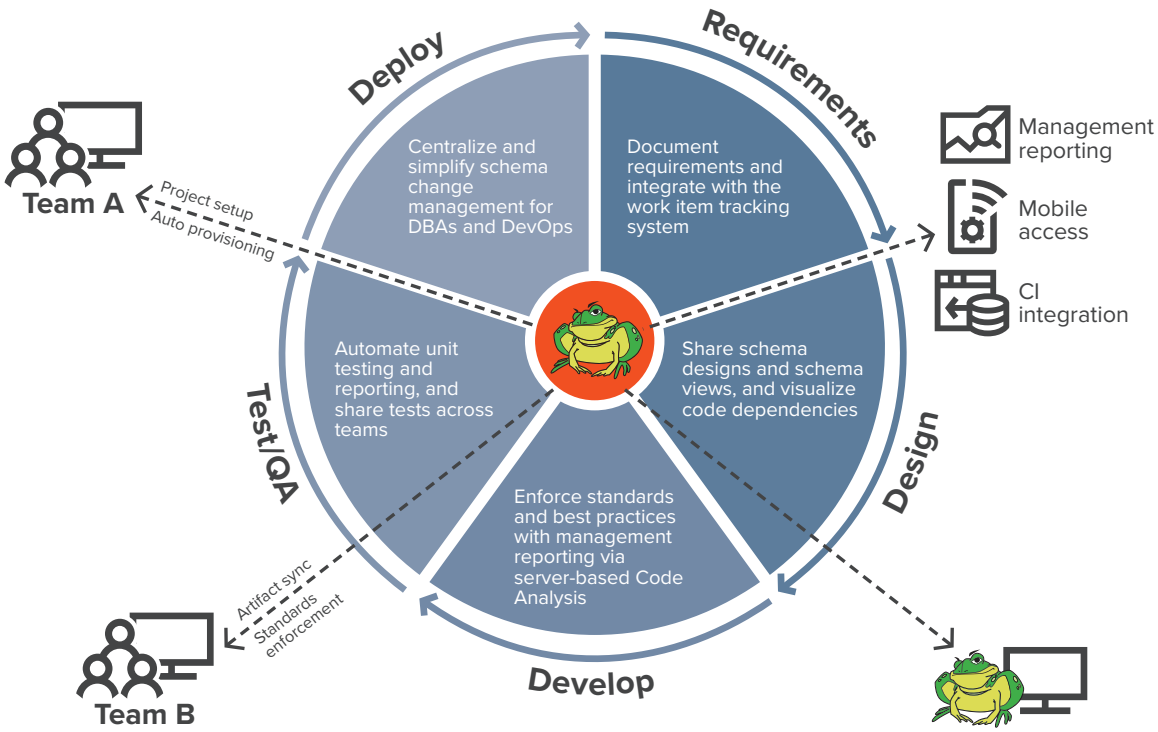


*Figure 1. Toad Intelligence Central is the hub of the database development lifecycle.*

Many windows in Toad Development Suite for Oracle can directly publish to and pull from Toad Intelligence Central.



Quest

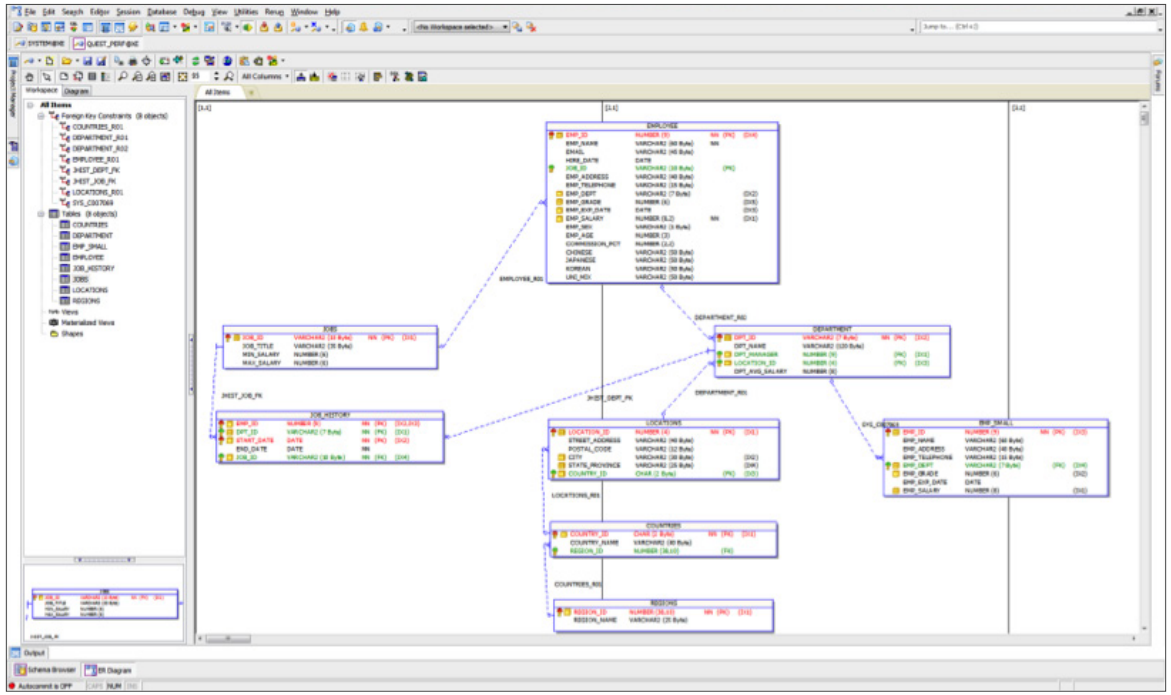

*Figure 2. Toad's ER Diagram makes it easy to visualize table relationships.*

## MANAGING PROJECTS

With Toad's Project Manager, you can create project folders and associate Toad files, database object references, web page links, and Windows folder and file references with the project, providing a consistent representation for all team members. With a single click, you can publish the asset list to the Toad Intelligence Central server, and any developer working on the project can pull the file to their Toad desktop. Each reference to an asset in the file is merely a pointer, so opening it in Project Manager accesses the latest version of the database object or file.

## UNDERSTANDING DATABASE RELATIONSHIPS AND CODE DEPENDENCIES

Toad's ER Diagram and Code Road Map enable developers to visualize graphically what tables and code relationships look like, view an object's properties and even make changes on the fly (see Figure 2).

# 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 modified and published to Toad Intelligence Central for team distribution.

# Testing code

## DEFINING TEST CASES

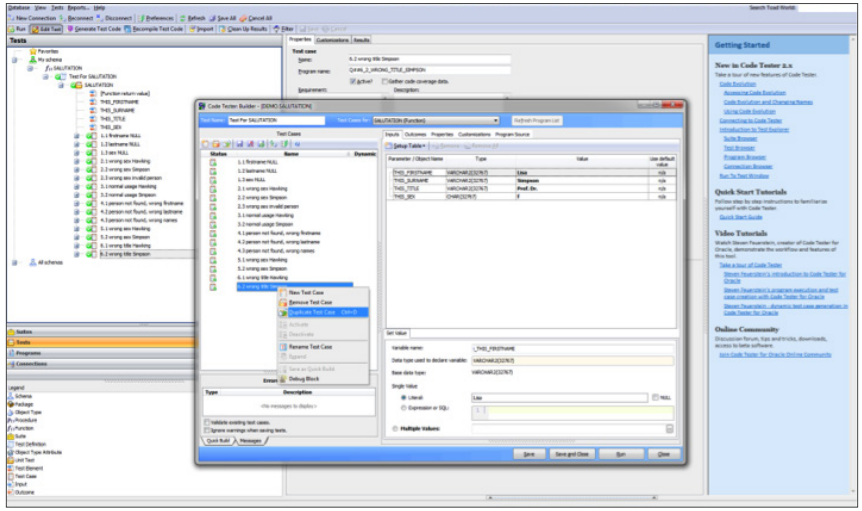

*Figure 3. Code Tester for Oracle automatically generates test code based on your use cases.*

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 Development Suite 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 3). 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.

The Code Tester for Oracle component of the Toad Development Suite 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.

# Debugging code

Of course, some test cases will fail, and you need to pinpoint the bugs. Toad Development Suite for Oracle supports a number of different debugging options, including the following:

- The Toad Editor supports debugging using the Oracle DBMS_DEBUG package as well as its own proprietary debugger for SQL*Plus scripts.
- Toad also supports debugging PL/SQL code using DBMS_OUTPUT statements and can automatically insert or remove them.
- External debugging allows an external application (such as Java or Oracle Forms) to call a PL/SQL stored procedure and pass any input parameters to a local debugging window in Toad automatically. This saves the developer from having to know what the input parameters are for a given use case

# 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 4, or they can view a summary of coding violations across a schema or project. Code Analysis can automatically publish code review results to Toad Intelligence Central as code reviews are conducted, so you can view historical information or trends.

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.

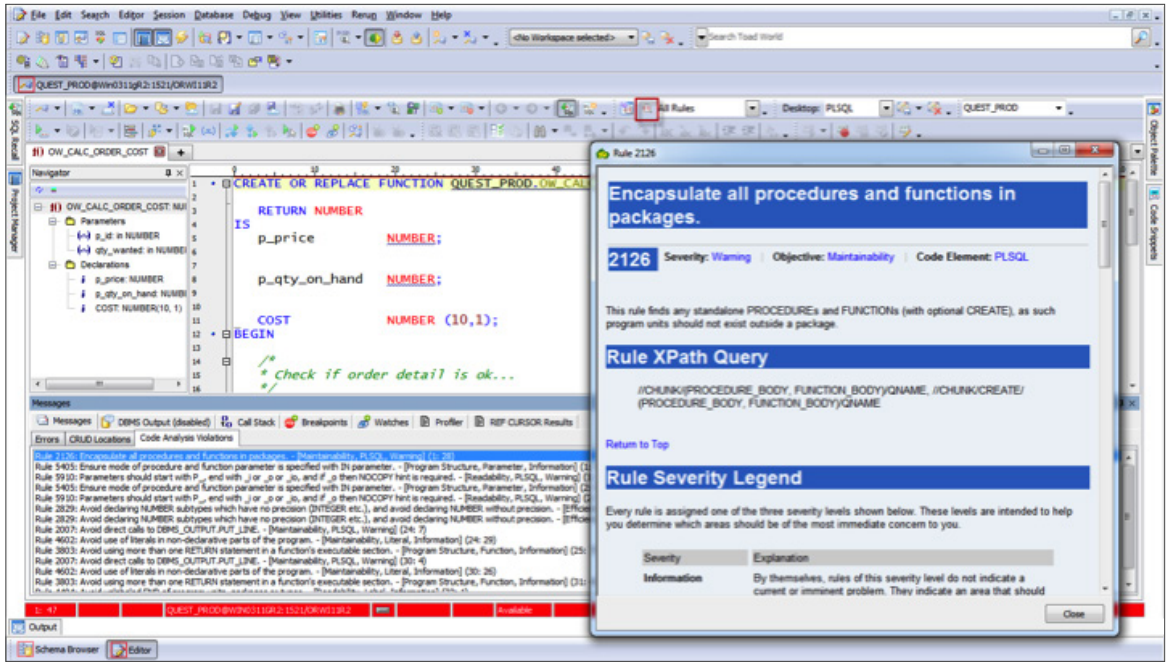


*Figure 4. Database developers can see and correct coding violations right in the Toad Editor.*



Quest

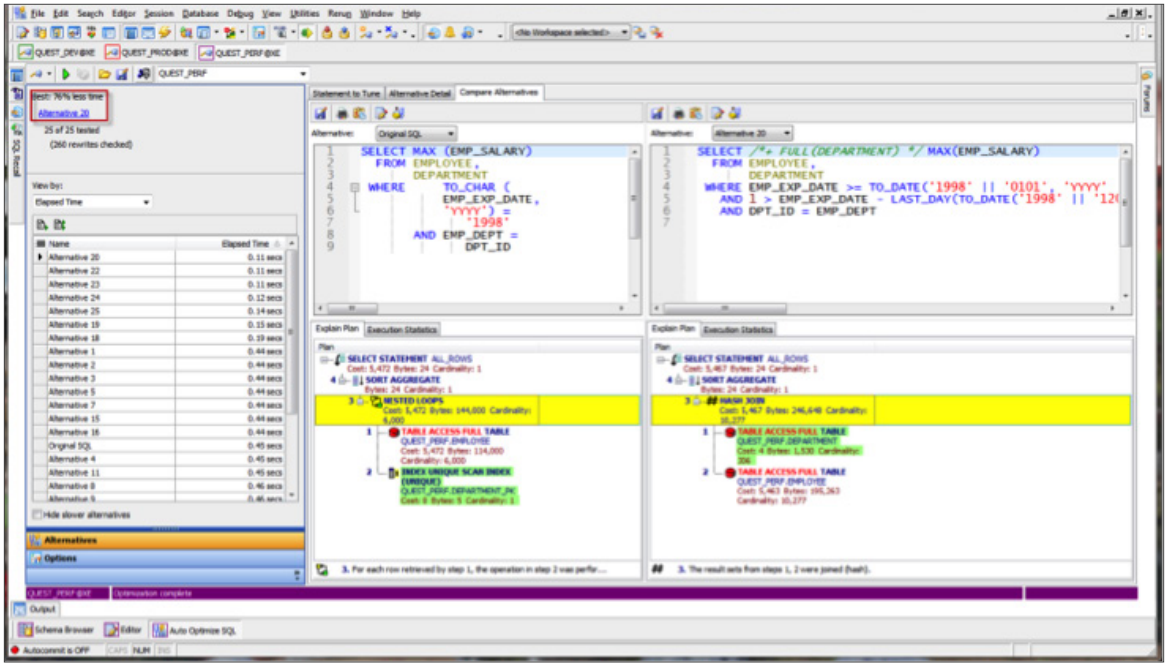


*Figure 5. Toad's Auto Optimize SQL is the most effective way for developers to tune PL/SQL and SQL.*

# 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 5). 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.

# Integration testing

## PL/SQL CODE REVIEW AND SQL SCAN

Toad enables you to execute Code Analysis and Scan SQL for multiple code units and store the results in Toad Intelligence Central, so you can perform code reviews over the entire project.

Code review automations can be created in Toad's Automation Designer and published to Toad Intelligence Central. These automations can be executed, scheduled or called from an external continuous integration process, such as Jenkins.

## PL/SQL REGRESSION TESTING

Agile development involves frequent updates to code, so automating the process of regression testing is critical. With Toad, you can easily run Code Tester over the entire code for a project and see whether it passed the tests.

Like code review automations, unit testing automations can be created in Toad's Automation Designer; published to Toad Intelligence Central; and executed, scheduled or called from an external continuous integration process, such as Jenkins.

## 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 Development Suite 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.

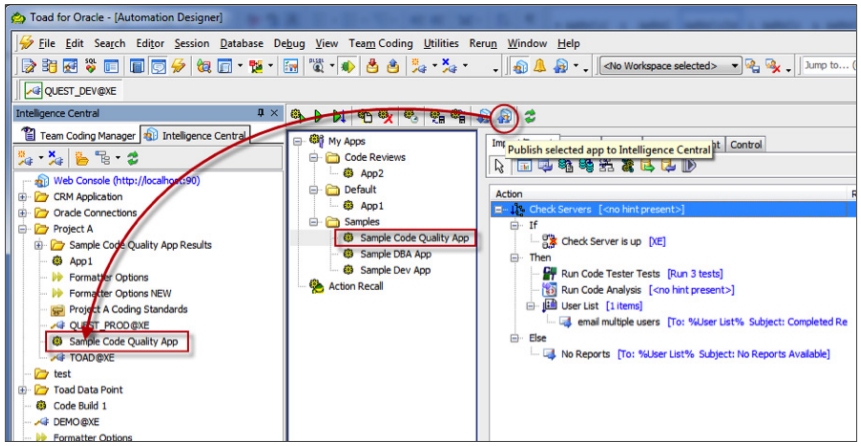


*Figure 6. With Automation Designer, you can easily create a workflow that you can reuse and share with others.*

# Automation

As we have seen, Toad helps you automate many steps in the development lifecycle and thereby reduce errors, save time and improve consistency. But Toad also offers a broader automation function. Using Automation Designer, you can save the state of a functional window (such as PL/SQL unit test executions, Code Analysis or Schema Compare), along with all its settings, as an action so it can be easily reused and shared with others. Actions can be grouped together to form apps, which can include conditional logic and other functions to represent repeatable, automated workflow, as shown in Figure 6. Both actions and apps can be published to Toad Intelligence Central and called as part of an external process such as continuous integration.

Part 2 of this e-book shows how you can take advantage of automation in Toad to bring your database development into parity with your agile application development.

PART 2

# Implement the automation components of Toad Development Suite for Oracle

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 Team Coding, Code Analysis, SQL Optimizer and Code Tester, components of Toad Development Suite for Oracle. The components touch four areas of interest in automating database development:

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

# Setting up the environment

As depicted in Figure 7, the implementation of Toad Development Suite for Oracle includes Toad for Oracle running on development staff desktops. It also calls for various Toad repositories and integration with version control systems.

Toad Intelligence Central runs on a Windows server that all developers using Toad can access. (See Part 3 of this e-book for more details on setting up Intelligence Central.).

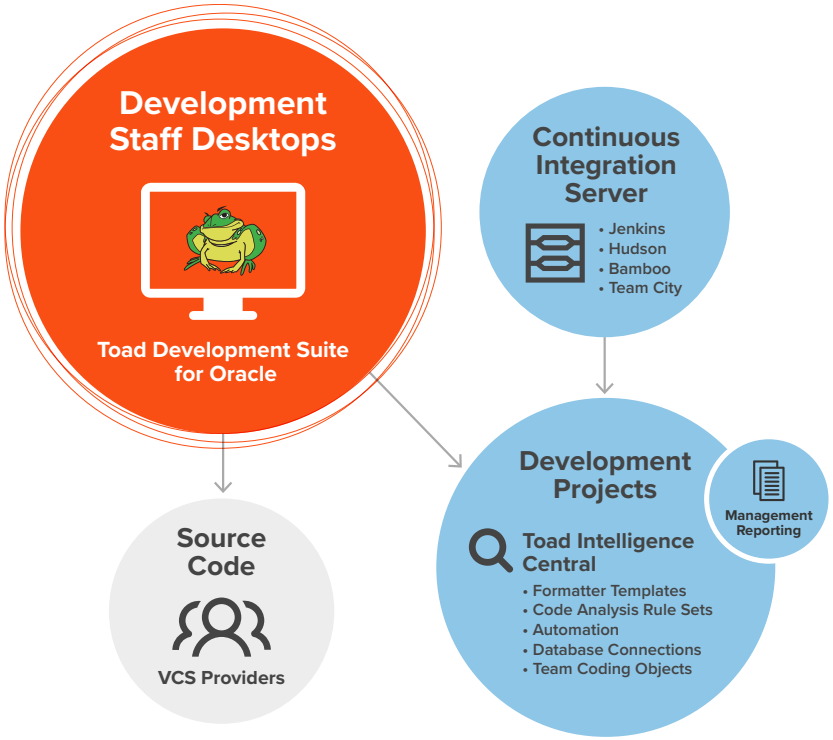


*Figure 7: Relationships among databases, version control and Toad products*

# Installing the developer desktop

First, you install Toad Development Suite for Oracle on development systems. 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 MySQL Freeware

> Toad Development Suite for Oracle is flexible enough to 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 Development Suite for Oracle is flexible enough to 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 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
- Microsoft Visual Studio Team Services (from Toad 12.9)
- Mercurial (from Toad 12.10)
- IBM Rational ClearCase
- Perforce
- CVS
- Git
- Subversion (CollabNet recommended)

Normally, Team Coding requires a set of repository tables to be installed in a schema in the Oracle database, but starting with Toad for Oracle v12.10, you can have the Team Coding objects installed to the Toad Intelligence Central server. The result is a truly centralized repository for all your projects without the need to install Team Coding on each database instance.

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 8, 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

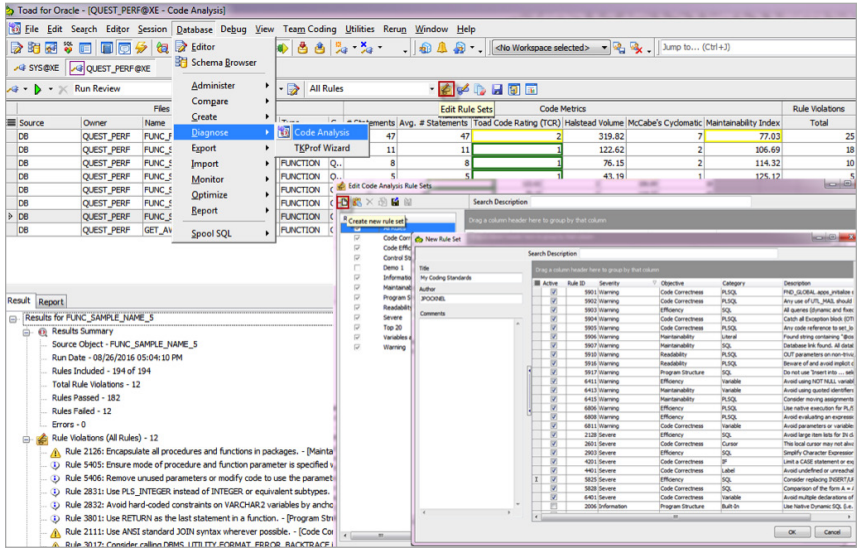

*Figure 8: Creating a new Rule Set in Code Analysis and selecting rules from the library*

levels from being checked in until the problems are fixed. For more information on setting Team Coding policies, watch this video.

Another option is to publish the results of code reviews to Toad Intelligence Central, which presents historical trending and event-based information through its web server.

## OPTIMIZING SQL AND PL/SQL

When a PL/SQL program runs slowly, you try to pinpoint the bottleneck. If the problem is a poorly written SQL statement, you then try to improve it. 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 9. 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 10) 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.

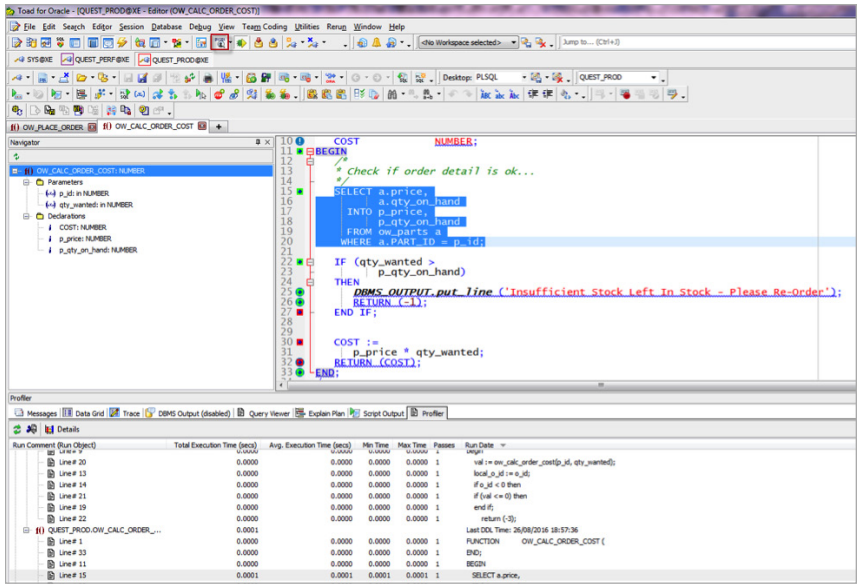

*Figure 9: PL/SQL Profiler identifying a slow-running SQL statement inside a PL/SQL program*

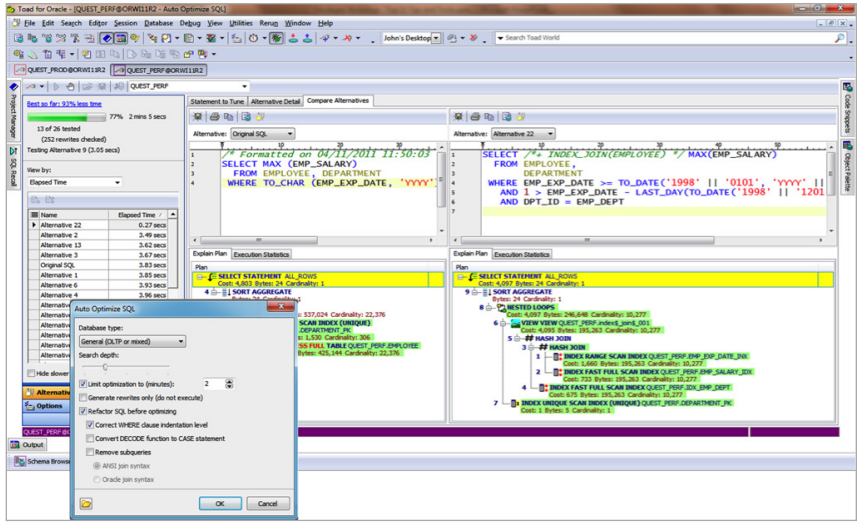

*Figure 10: Auto Optimize SQL automatically finding a faster SQL alternative*

# 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 stores execution data from unit tests in the repository 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.

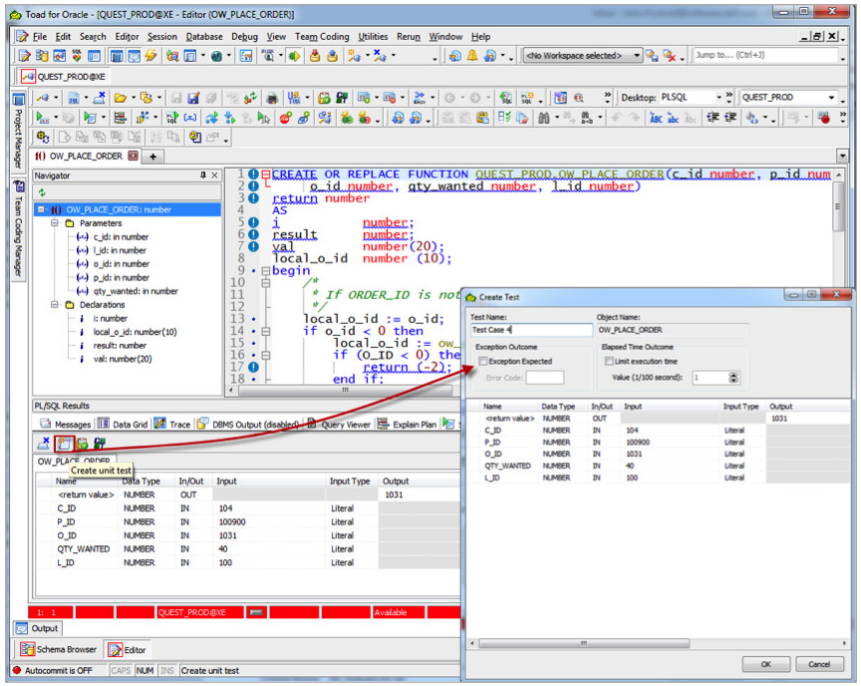


*Figure 11: Creating a PL/SQL unit test in Toad*

## CREATING UNIT TESTS IN TOAD EDITOR

As shown in Figure 11, 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.



Quest

## CREATING UNIT TESTS IN CODE TESTER

In Code Tester, the simplest way to create unit tests is to use the Run-to-Test feature, as shown in Figure 12. 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 13).

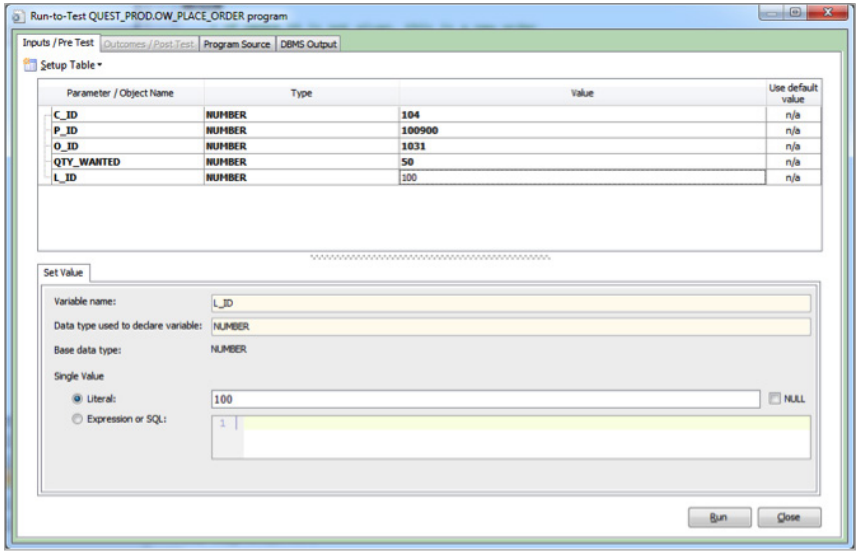


*Figure 12: Creating a PL/SQL unit test in Code Tester using Run-to-Test*

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.

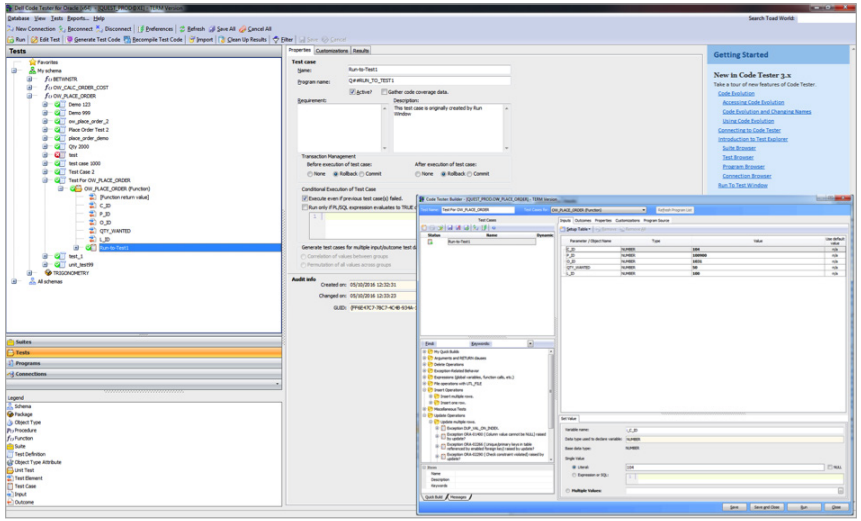

*Figure 13: Code Tester main window showing Test Builder (inset)*



Quest

PART 3

# Set up Toad Intelligence Central to maintain coding standards and access continuous integration servers

Shared resources and continuous integration 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 Intelligence Central (or simply "Intelligence Central") server, the centralized repository for making the database development tasks you perform in Toad for Oracle available for continuous integration. Once you have set up Toad Intelligence Central, you'll turn your attention to three areas:

- Maintaining coding standards (Code Analysis Rule Sets, format, PL/SQL) using Toad Intelligence Central
- Hosting and executing automated tasks (unit test execution, code reviews, code quality reporting) on Toad Intelligence Central
- Calling automated tests and code reviews from a continuous integration (CI) server



Quest

# Setting up the environment

As depicted in Figure 7, the implementation of Toad Development Suite for Oracle includes Toad for Oracle running on development staff desktops. It also calls for various Toad repositories and integration with version control systems.

Install Toad Intelligence Central on any Windows server that all your developers using Toad can access.

# Toad Intelligence Central (Community Edition) server installation

Toad for Oracle users can install Toad Intelligence Central (Community Edition), which is free of charge. There is also a paid-for version of Toad Intelligence Central, with data connectivity. Analysts can use it with Toad Data Point and more than 50 different types of data sources.

The Toad Intelligence Central Deployment Guide contains all the information you need to install and configure the Intelligence Central server and add new users and groups.

As shown in Figure 14, Intelligence Central consists of the server, the Web Server and the Admin Console.

Once you have created users and groups in Intelligence Central, you can connect Toad for Oracle to your Intelligence Central instance (see Figure 15) and begin publishing files.

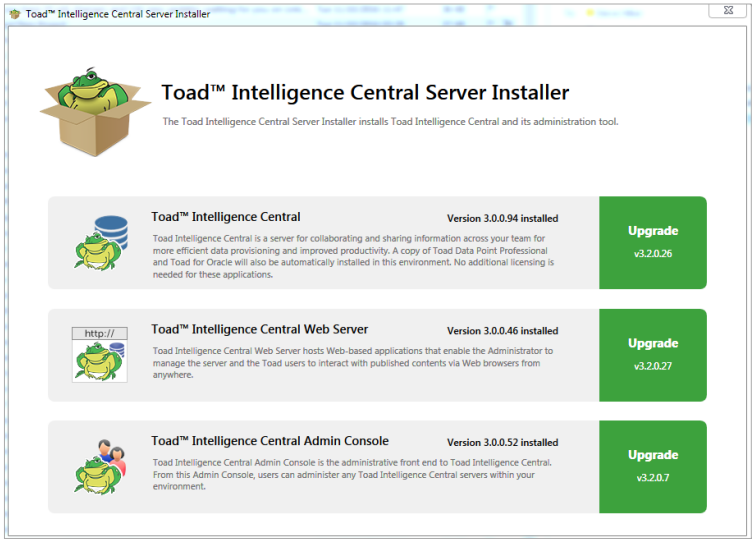


*Figure 14: The Toad Intelligence Central installer wizard*

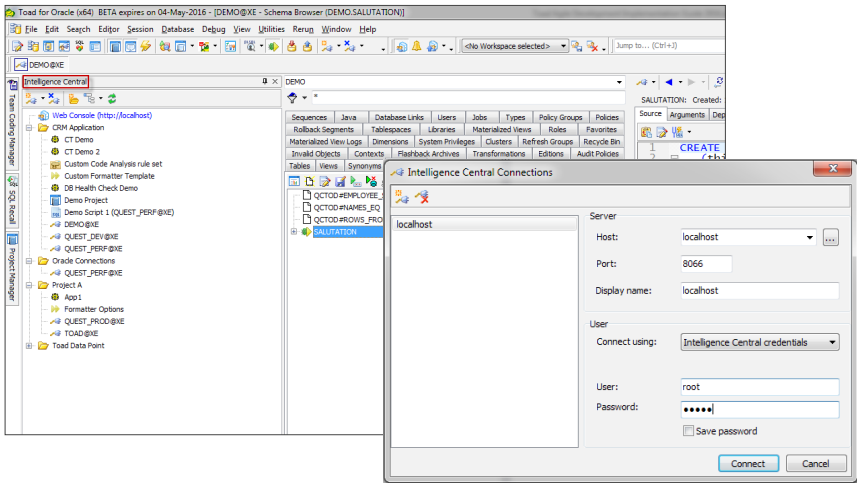


*Figure 15: Connecting Toad for Oracle to the Toad Intelligence Central server*



Quest

To run automated Toad tasks, Toad for Oracle will be installed automatically as part of the Intelligence Central installation. The implementation also requires an Oracle client for Windows running on the same computer as Intelligence Central. The client should match the Oracle server in version number and bit-width (32 vs. 64).

# Maintaining coding standards using Toad Intelligence Central

The following sections describe the different types of artifacts (automation, Rule Sets, templates, files, etc.) that can be published to and pulled from the Intelligence Central server. Developer teams should nominate an individual to be responsible for maintaining, updating and publishing the artifacts to the appropriate project folder on the Intelligence Central server.

Toad for Oracle users who are connected to the Intelligence Central server and have access to the artifacts will receive notification of changes. They can then pull them from Intelligence Central to their local Toad application. The updated versions overwrite existing ones to ensure the whole team is working to the same standards.

## CODE ANALYSIS RULE SETS

We covered the creation of Code Analysis Rule Sets in Chapter 2 of this e-book. You can publish those Code Analysis Rules and Rule Sets to the Toad Intelligence Central server. Easy sharing helps ensure that all developers on your team are reviewing their code to the same standards.

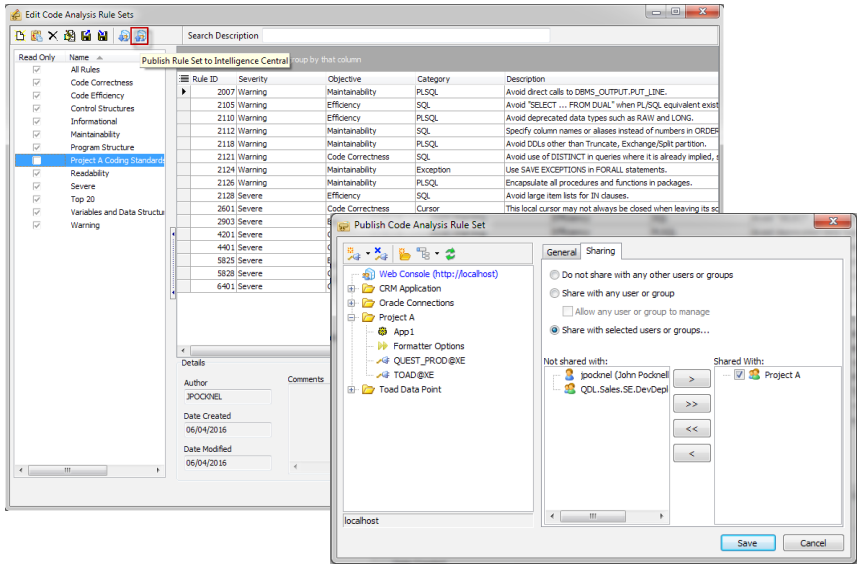


*Figure 16: Publishing the new Rule Set to the Toad Intelligence Central server for sharing with Project Team A*

To publish your new Rule Set to the Intelligence Central server, click the "Publish Rule Set to Intelligence Central" button, as shown in Figure 16.

If you have customized Rules, you can also save and export them. Note that Intelligence Central saves those separately (as XML files) and keeps them apart from Toad's native rules.

## FORMATTER TEMPLATES

Formatter is a feature accessible from the Editor. It applies consistent formatting to make code easier to read and maintain, and to keep it in compliance with your organization's standards.

You can modify the default Formatter template file (FmtPlus.opt) by selecting View | Toad Options | Formatter. To make it available to the

entire project team, save it to the Toad Intelligence Central server as described in “Code Analysis Rule Sets” above.

## NEW PL/SQL TEMPLATES

New PL/SQL Templates enable developers to create new PL/SQL objects based on a set of standard code constructs (templates).

Again, you can modify the default PL/SQL template files by selecting View | Toad Options | PL/SQL Templates, then save them to the Toad Intelligence Central server as described in “Code Analysis Rule Sets” above to ensure all developers write code consistently.

To publish templates to the Intelligence Central server, locate the files named `Newtemplate_name.sql` in your Toad User Files directory. Select Add | Folder Items to store them in Project Manager.

New PL/SQL Templates enable developers to create new PL/SQL objects based on a set of standard code constructs.

# Hosting and executing automated tasks on Toad Intelligence Central

The real power in the combination of Toad Intelligence Central and Toad for Oracle lies in the ability to store and execute automated tasks, such as PL/SQL unit tests and code reviews. Continuous integration servers like Jenkins, Hudson, Bamboo and Team City can access those tasks and call them from Intelligence Central as part of an automated build process.

## AUTOMATING PL/SQL UNIT TEST EXECUTION

To automate the execution of your PL/SQL unit tests, first create a Code Tester Action.

Open the Automation Designer window and double-click the "Code Tester" button on the "DB Misc" tab to create a blank Action called "Code Tester 1." Right-click the Action and rename it to suit your needs. On the Properties page, you can specify that you wish to publish the results of your unit tests to Toad Intelligence Central.

Next, select the Test Cases, Unit Tests or Test Suites you want to run by clicking the "plus" symbol button in the toolbar and selecting from the Code Tester repository, as shown in Figure 17. You can expand the Unit Tests to show all the Test Cases contained within them. Specify an output directory to which the test results file (text or XML format) will be written.

Follow these steps so that PL/SQL unit tests can be called externally, either directly through command line or from a continuous integration server as part of an automated build:

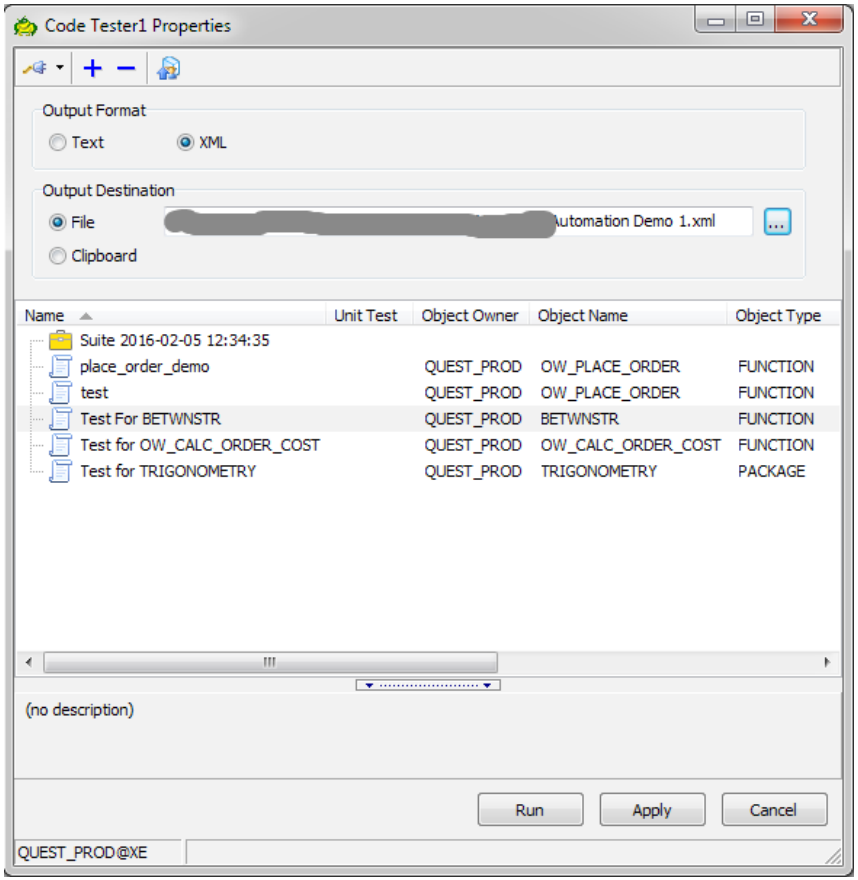


*Figure 17: Properties of a Code Tester App, showing unit tests to be executed*

1. Enable publishing of the unit test execution results to the Intelligence Central server by clicking the "Publish results to Intelligence Central" button on the Code Tester Action toolbar.

2. Publish the Action to the Intelligence Central server by clicking the "Publish to Intelligence Central" button on the Automation Designer toolbar.

The Publish dialog will open, as shown in Figure 18. Select the project folder to which you want to publish and the users and groups with whom you want to share the Action.

## AUTOMATING CODE REVIEWS

Similar to the procedure for Automating PL/SQL unit test execution, you automate the execution of your code reviews by first creating a Code Analysis Action.

Open the Automation Designer window and double-click the "Code Analysis" button on the DB Misc tab to create a blank Action called "Code Analysis 1." Right-click the Action and rename it to suit your needs. On the Properties page, you can specify whether to simply run a code review or to include a SQL Scan. You can also specify which Code Analysis Rule Set to use and whether you wish to publish the review results to Toad Intelligence Central.

Next, select the PL/SQL code to review, whether from the Windows file system or from the Oracle database. Specify an output directory and names for the output files, as shown in Figure 19.

Follow these steps so that PL/SQL code reviews can be called externally, either directly through command line or from a continuous integration server as part of an automated build:

1. Enable publishing of the unit test execution results to the Intelligence Central server by clicking the "Publish results to Intelligence Central" button on the Code Analysis Action toolbar.
2. Publish the Action to the Intelligence Central server by clicking the "Publish to Intelligence Central" button on the Automation Designer toolbar.

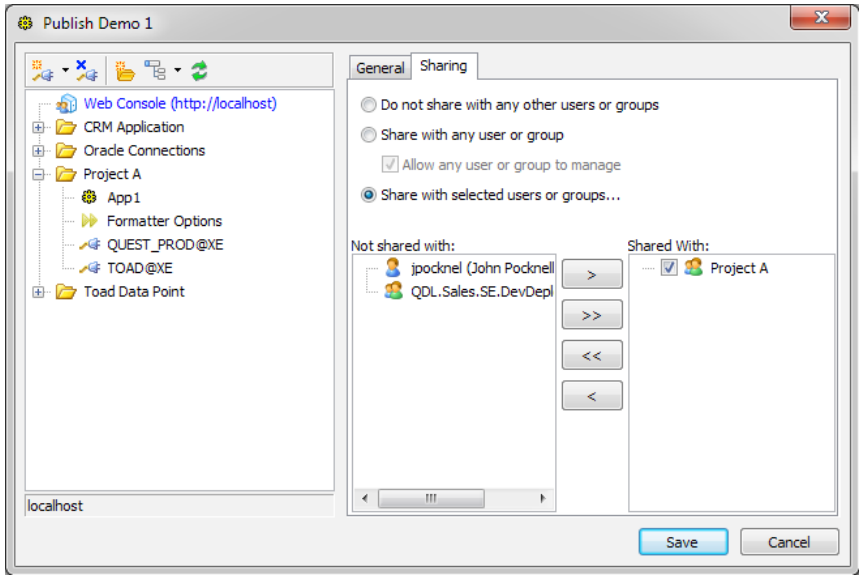


*Figure 18: Publishing a Code Tester Action to the Toad Intelligence Central server and setting access rights*

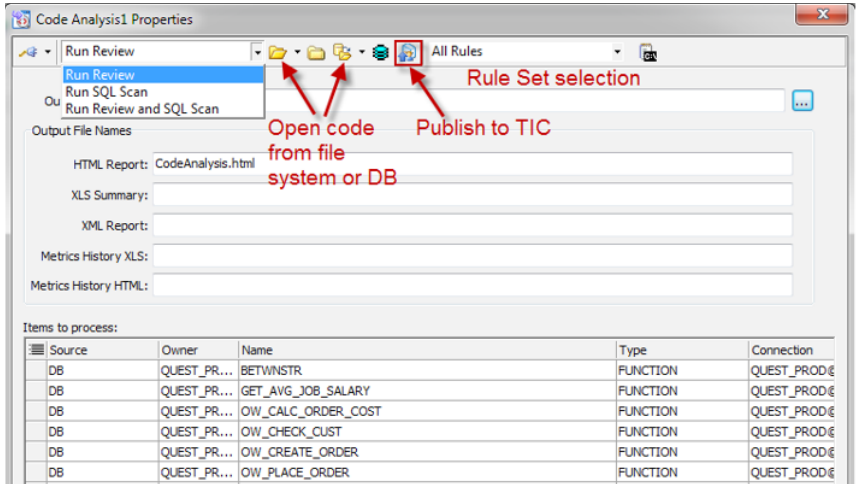


*Figure 19: Creating a Code Analysis Automation Action using Automation Designer*

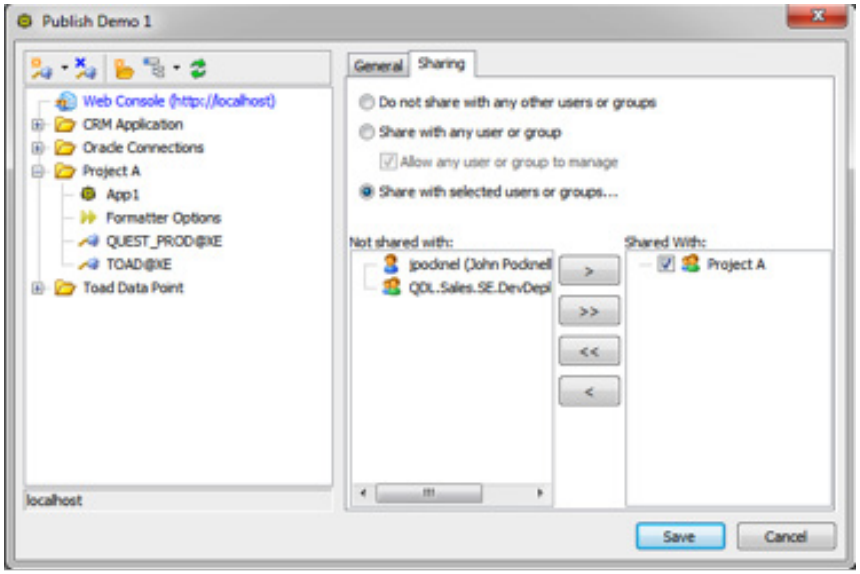


*Figure 20: Publishing a Code Analysis Action to the Toad Intelligence Central server and setting access rights*

The Publish dialog will open, as shown in Figure 20. Select the project folder to which you want to publish and the users and groups with whom you want to share the Action. Watch this video for more information on publishing standards and automation to Intelligence Central.

## REPORTING ON CODE QUALITY USING THE TOAD INTELLIGENCE CENTRAL WEB SERVER

Once you start publishing Toad for Oracle automated tasks to Intelligence Central, you'll see that, whether you execute them from Intelligence Central or remotely, the results will be stored in Intelligence Central, available for viewing through the web server. Access the server by clicking the hyperlink in the Intelligence Central fly-out window in Toad. Once you are signed in, you will see a home page similar to Figure 21.

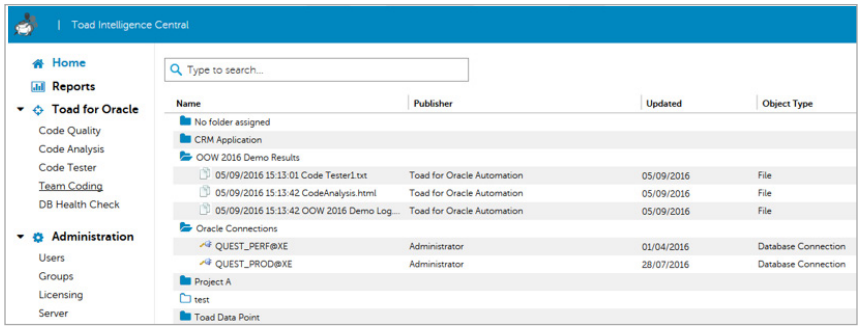


*Figure 21: Toad Intelligence Centrtal web server home page*

Through the web server you can examine multiple reports that offer insights into code quality across a given database or schema. As shown in Figure 22, the highest-level report combines PL/SQL unit test (in the Code Tester portion of the dashboard), code review (in the Toad Code Rating portion) and analysis metrics with event-based data from Team Coding.


*Figure 22: Code Quality dashboard showing data from PL/SQL unit tests, code reviews and events from Team Coding*

# Calling automated tests and code reviews from a continuous integration (CI) server

You can include automated PL/SQL unit test executions (regression testing) and code reviews as part of an automated build process hosted on a CI server, such as Jenkins, Hudson, Bamboo or Team City. An execution URL combined with an HTTP POST method tells the CI server what to execute on the Toad Intelligence Central server.

After you have published your Toad application to Intelligence Central, right-click it in the Intelligence Central pane and select Server | Get execution URL, as shown in Figure 23. Once the CI server receives the URL, it will include the contents of the Toad automation as part of its build. Watch this video for more information on running tasks on Intelligence Central and calling tasks from a CI server.

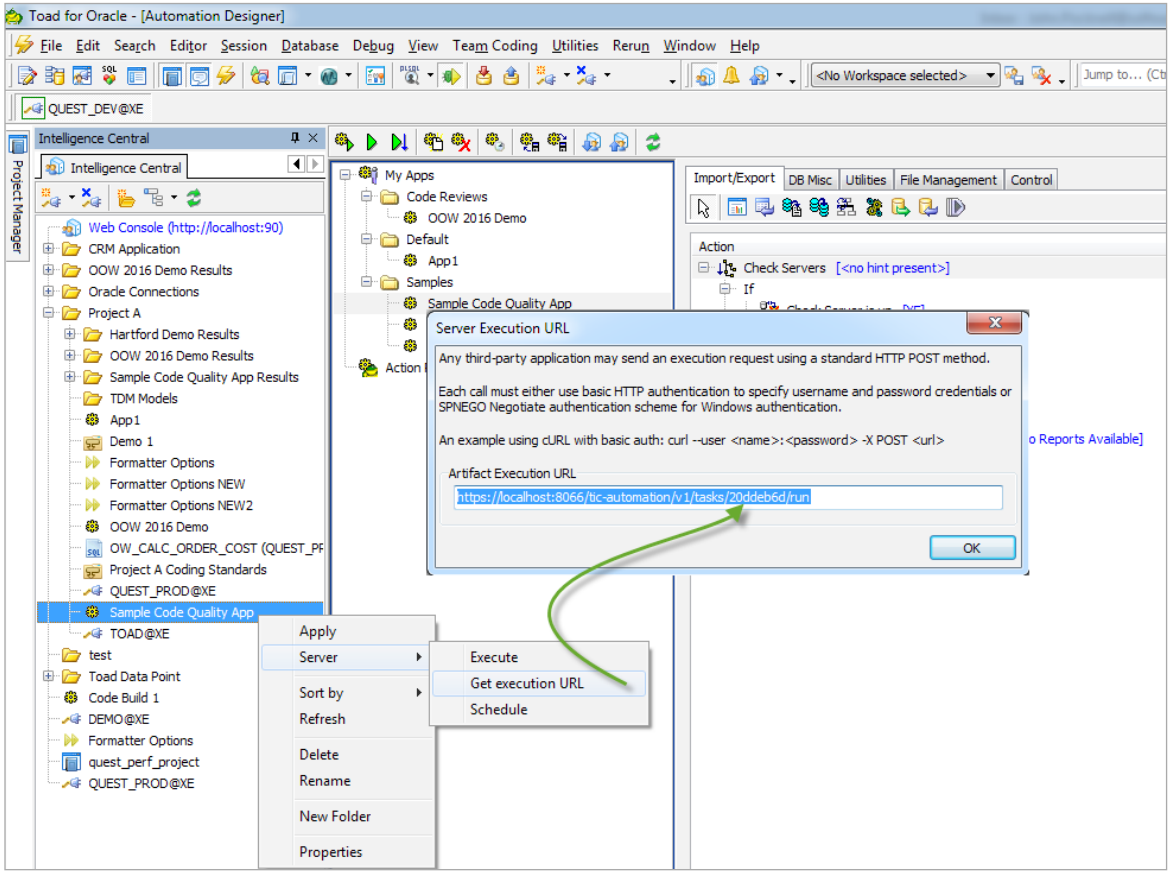


*Figure 23: Generating the execution URL to call a Toad automated PL/SQL regression test and code review*



Quest

# 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 Development Suite for Oracle and Toad Intelligence Central deliver automated features like Team Coding, Code Analysis, PL/SQL Profiler, SQL Optimizer and Code Tester. Intelligence Central also stores artifacts like automation, Rule Sets and templates in a repository for use in automated build processes on a continuous integration server.

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.

Quest

## ABOUT QUEST

Quest helps our customers reduce tedious administration tasks so they can focus on the innovation necessary for their businesses to grow. Quest® solutions are scalable, affordable and simple-to-use, and they deliver unmatched efficiency and productivity. Combined with Quest's invitation to the global community to be a part of its innovation, as well as our firm commitment to ensuring customer satisfaction, Quest will continue to accelerate the delivery of the most comprehensive solutions for Azure cloud management, SaaS, security, workforce mobility and data-driven insight.

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.



Ebook-EnablingAgileDBDev-US-GM-26524