

VIII Konferencja PLOUG
Kościelisko
Październik 2002

Quest Software Tools supporting Developer Teams

referat sponsorski

Doug Daniels & Darin Pendergraft

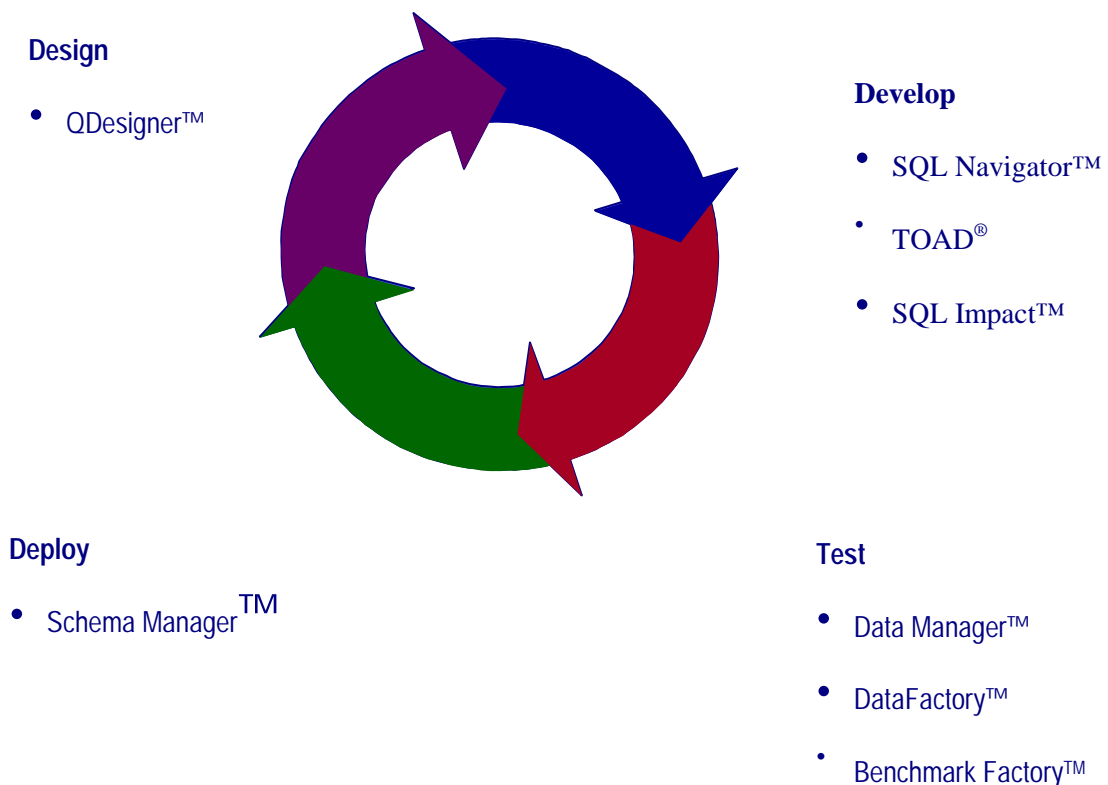
Quest Software

1. TOAD Suite – powerful tool for development

1.1. Introduction

The dot-com bubble bore witness to many IT shops forgetful of simple business metrics such as assets, liabilities, and the bottom line. The subsequent shakeout has had an effect on even staid, old-line economy data centers. Today, the current economic environment demands IT organizations get the most out of both their skilled technical professionals and their existing IT infrastructure. In these times of lean budgets, IT managers and CIO's must be sensitive to issues of return on investment (ROI), total cost of ownership (TCO), IT asset performance, and employee productivity. Leveraging existing IT assets and human capital with easy-to-use, affordable tools is the mission of the TOAD® Suite and Quest Software. In designing the TOAD Suite, Quest has focused on three key business fundamentals common to all Oracle RDBMS development shops – assets, liabilities, and the bottom line. All three of these business metrics bear great importance to an IT organization's ability to execute its mission of supporting the business with robust, mission critical applications throughout the application development lifecycle. In today's economy, only those IT organizations that pursue RDBMS application development with an eye toward business results are those that will succeed in the global marketplace.

Let's examine each of these business drivers in turn and how the TOAD Suite can dramatically improve them in any Oracle development shop.



1.2. Assets

Quest defines IT assets as all the databases, code, developers, and applications that provide real value throughout an organization's development lifecycle. New application design, custom tuned, debugged, and formatted code, and developer productivity all represent important competitive differentiators to the firm. These assets should be recognized as such and brought to bear on the firm's bottom line.

New business applications must be architected correctly, efficiently, and on time to meet the competitive challenges of the global information-based economy. Correctly designing and migrating RDBMS applications will provide the firm with a competitive advantage for years to come, however the cost of custom development represents a large capital investment in human capital and other assets. Providing application designers with best-of-breed solutions such as **QDesigner™**, allows designers to combine object-oriented, conceptual and physical data object modeling capabilities in a single, integrated environment. With an intuitive user interface and support for more than 30 of today's most popular RDBMS's, this tool speeds up software design and analysis - helping designers and developers deliver increasingly complex distributed applications on time. Properly architecting applications in the beginning of the lifecycle will greatly decrease the cost of the application over its lifetime.

As the lifecycle moves to the Development stage, custom code represents an increasingly important organizational asset as it is unique to your organization and, if implemented properly, represents an important advantage over your competitors. Something as simple as correct code formatting and compilation can save valuable development cycles. Through the use of tools such as **Formatter Plus™**, an IT shop can be sure that its code is formatted to the organizational standard for quick and clean readability and debugging. In conjunction with the **TOAD Debugger**, developers can then step through their formatted code line-by-line as it executes, easily setting conditional breakpoints, viewing watch variables, and viewing the call stack. Maintaining correctly formatted and compiled code thus allows a greater return on your code assets.

Once the custom application is developed, it must perform. If the custom code running your critical applications isn't tuned properly, it can be considered a non-performing or under-performing asset. Using tools such as the **Quest Xpert Tuning module** to make observations about a selected SQL statement and the underlying database environment, developers can easily and quickly tune critical application components. Automating the process even further, **the Xpert Tuning module** generates the tuned code (SQL rewrites) that offer the highest likelihood of improving the SQL statement, thus allowing the developer to merely select the best tuning solution based on the wealth of Oracle expertise and experience that resides within the tool. Such automation of a typically time consuming and tedious task provides real ROI for the IT shop.

Finally, developers and designers represent valuable human capital that should be put to optimal use across the lifecycle spectrum. Since the Oracle environment is large and complex, even the most senior of technical professionals will have the need for detailed code samples, function definitions, and error codes. Instead of wasting valuable development cycles researching arcane code syntax, IT managers can empower their development team with out-of-the-box knowledge bases that can answer most PL/SQL development issues. Using award-winning knowledge bases such as the **Knowledge Xpert™ for PL/SQL**, new and expert developers alike can quickly gain an expert understanding of their databases and code structures. By employing expensive human capital more efficiently, tools such as the Quest Xpert Series provide a real lowered Total Cost of Ownership (TCO) for firms.

1.3. Liabilities

Quest defines IT Liabilities as all the databases, code, developers, and applications that provide real or potential claims or obligations on an organization's resources across the application lifecycle. Such claims might be the cost of lost business due to an application's failure or the loss of employee productivity due to disconnects between the development team and the DBA team. Thus, in order to understand the real cost of business in today's IT shop, one must be aware of the real and potential claims and responsibilities that rest on the Data Center's resources.

A primary responsibility of Data Centers is to provide application availability, even under maximum load in a production environment. Many organizations never load test their critical database servers or do so only with dummy test data or after the application has been deployed. Thus when peak load is encountered and the application fails or substantially degrades, many organizations are left with long service outages and frantic problem solving. Using world-class tools such as **Benchmark Factory® for Oracle**, developers and administrators can simulate thousands of users accessing their applications in a pre-production state. This allows them to safely determine system capacity, pinpoint system bottlenecks and isolate system stress related issues *before* the problems occur in their production environment. Furthermore, tools such as the Quest **DataFactory®**, provide for data generation of millions of rows of meaningful, syntactically correct test data to ensure that organizations can be certain their load tests reflect 'real' user data transactions. Indeed, for 24/7 mission critical environments, load testing with meaningful data is an essential way to minimize the liability of an application's unexpected degradation. Meaningful load testing thus provides decreased liability of critical applications failing when they are most needed.

Human capital represents a liability as well as an asset. Database administration and maintenance typically consumes vast resources in a complex, enterprise-wide IT shop.

Managing users, rollback segments, maintenance cron jobs, and other associated tasks often requires the time of a full time and experienced staff. To make matters worse, many PL/SQL developers are not DBA's, and thus there is an often alarming disconnect between the developer's concerns and the responsibilities of the DBA. Using a single interface tool such as **TOAD** and the **Quest DBA Module**, developers and DBA's alike will have a single tool and interface. Cross-pollination inevitably occurs and the line between the folks who understand your code (the developers) and the folks who understand your infrastructure (the DBAs) blurs.

Such a best-of-breed single interface into the different worlds of database development and database management allows greater visibility of both teams across critical business application development cycles. Savings in time and worker productivity are only the beginning of the benefits that can be achieved using TOAD in conjunction with the DBA Module.

1.4. Customers Speak

"TOAD enables the developer to concentrate upon the development cycle and not focus upon learning complex syntax and tasks that require a high level of Oracle experience. "

"PL/SQL debugging and SQL tuning are complex tasks, even with the aid of TOAD. Without TOAD, they become specialized tasks. "

- IT Manager at large insurance conglomerate

Finally, DBAs are expensive and require a substantial long-term investment in training and on-going employee development costs – a major liability in any IT organization. Leveraging existing employees with minimal Oracle expertise is one method IT shops use to get the most bang for their buck out of technical professionals. In utilizing best-of-class knowledge bases such as **Knowledge Xpert for Oracle Administration**, IT managers can be sure that their DBA's have the tools to succeed. By decreasing the training and administration costs of the DBA staff, firms can achieve significant lowered TCO and reduced obligations on the technical staff's valuable time.

1.5. The Bottom Line

Assets and liabilities are the hallmark of performance metrics for most organizations, and data centers developing complex Oracle-based applications are no exception. Code embedded inside your critical custom business applications is one of the most important business assets in today's global economy. Proprietary database applications provide the enterprising firm better margins and faster response time to rapidly changing business conditions. Custom tweaks and tuning of applications allows your critical code to perform above and beyond your competitors.

Programmers and other technical professionals are expensive and represent liabilities or obligations against your organizations. IT managers must get the most out of their people while being sensitive to the demand for their valuable skills elsewhere. By giving technical people best-of-breed tools to do their job, an IT manager goes a long way in creating an efficient and rewarding environment for employees. Furthermore, leveraging increased productivity from your valuable human capital goes a long way toward increasing the firm's bottom line. Indeed, Alan Greenspan cites gains in worker productivity at the heart of America's long-running expansion.

Figure 1 depicts just a few of the bottom line benefits encountered by a large multi-national insurance conglomerate which applied the TOAD Suite to its RDBMS application development life-cycle. A quick glance at the numbers reveal that over 70% of the respondents found TOAD easier to use than Oracle's SQL*Plus editor. Furthermore, 70% of respondents report that the application development cycle has been shortened due to the TOAD Suite. Finally, nearly half of the respondents report savings in time or other resources due to the tuning and debugging features found inside the Suite. These numbers illustrate just some of the savings and return on investments enterprising firms encounter when using a powerful integrated development environment such as the TOAD Suite.

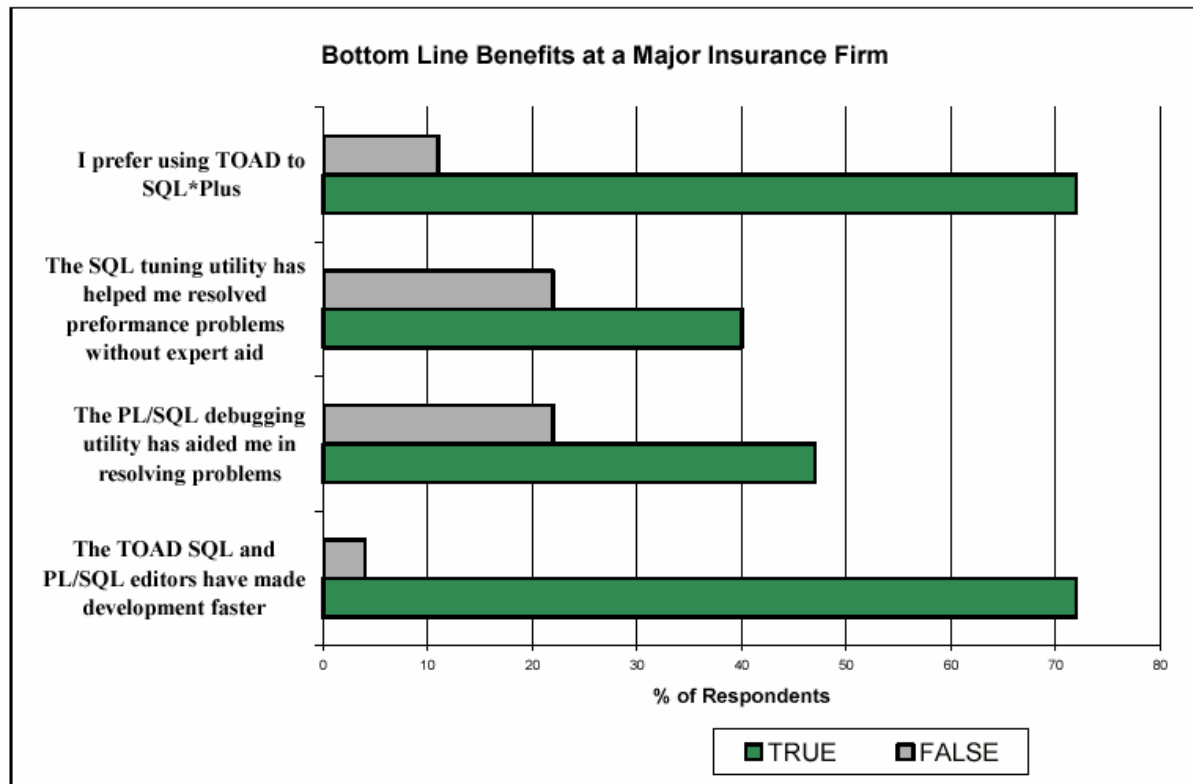


Figure 1. Benefits of the TOAD Suite on one firm's lifecycle

Thus, by bringing to bear an integrated suite of RDBMS development tools, IT shops around the world have recognized an increased return on their existing IT assets and an overall decrease in their real and potential liabilities. At its heart, the **TOAD Suite** consists of **TOAD**, the best-of-breed Oracle development tool known the world over for its intuitive database schema browser and associated tool sets. In addition to **TOAD**, Quest has integrated testing, tuning, administration, reference, and design tools to provide a complete solution to increase your bottom line in the data center. By providing a single, out-of-the-box, total lifecycle development platform for Oracle RDBMS applications, IT shops can realize immediate, tangible business gains.

1.6.The TOAD Suite Solution

The **TOAD Suite** has been developed as an easy to use, full life-cycle development platform for your critical Oracle database development projects. Using best-of-breed products such as **TOAD**, the Suite allows your development team to get started developing world-class applications immediately. Table 1 summarizes the components of the **TOAD Suite**.

Table 1. Composition of the TOAD Suite

TOAD Suite Components	
TOAD	Best-of-Breed tool provides quick and easy PL/SQL development and administration.
QDesigner	Provides integrated application analysis and design.
Knowledge Xpert for PL/SQL Development	A comprehensive online technical reference covering the entire lifecycle of PL/SQL programming with background information, best practices, syntax diagrams, and extensive code library.
Knowledge Xpert for Oracle Administration	A comprehensive online technical reference providing insights, syntax, examples and step by step procedures for all of a DBA's daily Oracle RDBMS tasks.
Benchmark Factory	Load testing and capacity planning for critical application environments.
Data Factory	Quickly populate test applications with meaningful test data.
Quest DBA Module	Provides a well-rounded environment for both DBA's and developers to handle common administration tasks — all within TOAD's intuitive interface.
Formatter Plus	Allows for analyzing and formatting entire PL/SQL applications including instant code reviews with recommendations.
Xpert Tuning Module	Supplements Oracle tuning skills for developers, stepping up their application tuning capabilities and application performance.
PL/SQL Debugger	Provides full-featured debugging environment to step through PL/SQL code one line at a time as it executes.

1.6. Conclusions

It is apparent that IT organizations must take a business view toward application development and deployment in the future. No longer can mission critical applications be thought of as merely overhead on the bottom line; indeed, in many ways, RDBMS-based business applications are increasingly *driving* the bottom line of large and medium sized organizations. Isn't it time we applied the same business principles and metrics to our application development process that we do to our balance sheets? Through Quest's easy-to-use, cost effective Oracle RDBMS development tools, organizations can immediately begin to feel the positive impact on real business issues.

2. Benchmark Basics

by Bernard Farrell and Brian Butler

2.1. Introduction

One can think of benchmarking as the running of a set of standard tests on one system to compare its performance and capacity to that of other systems. Benchmarks measure:

- Performance of an enterprise system
- Performance of a specific subsystem, such as disk, video, CPU, or memory
- Performance of a particular application, such as spreadsheet, database, or CAD/CAM

- Performance of server-based applications, such as database, file, messaging, or Web
- Capacity of an enterprise system

All good benchmarks use a well-defined testing methodology based on real-world use of a computer system. Benchmarks measure system performance in a deterministic and reproducible manner, allowing a user to judge the performance and capacity of their systems properly. When used properly, benchmarks determine:

- Reliability Bottlenecks
- Enterprise capacity
- Impartial purchasing information

2.2. Why Benchmark?

Reliable enterprise-server performance equates to excellent customer service. Excellent customer service equates to maximized profits. In a highly competitive and unforgiving e-commerce environment, system performance is the defining factor that keeps customers satisfied. Satisfied customers provide higher company revenue and profit. Implementing a well-planned benchmarking program for e-commerce is essential to staying competitive.

2.3. Benchmark Basic Components

Benchmarks are made up of the following basic components:

- Specifications
- Control logic
- Implementation **Specifications**

Benchmark specifications provides:

- Details
- Design goals
- Data points
- Execution plans

Details

Details are at the heart of benchmarking. The Transaction Processing Council (TPC) is a non-profit corporation that defines transaction processing and database benchmarks.

The TPC disseminates objective, verifiable performance specifications for the computer industry. For example, the TPC-C (V5.0) specification, that defines on-line transaction processing benchmarks, is a detailed document.

There is a trend toward standard industry benchmarks within software areas or domains. Groups of vendors define standard industry benchmarks for its domains. The Standard Performance Evaluation Corporation (SPEC) establishes, maintains, and endorses a standardized set of relevant benchmarks and metrics for performance evaluation of modern computer systems.

Design Goals

Well-defined design goals are essential for a successful benchmark. A successful benchmark must possess the following characteristics:

Scalability—Benchmarks, like enterprise systems, must be able to serve large numbers of users in a testing process without requiring major changes

Broad architectural scope—Enterprises have a wide range of operating systems and software applications to test. Benchmarks must be flexible and run concurrently on an enterprise system

Easy to use and understand—System administrators do not have extra hours to analyze complicated benchmarks. A benchmark must be easy to install, run, and understand

Representative of the system being tested—Benchmarks must accurately represent a system-under-test. For example, if a system-under-test is a database, the benchmark must reflect valid test data measurements, improved customer database access time, and other factors important to the overall operation of a database

Accuracy

Benchmark testing must be accurate, and reflect the demands placed on enterprise systems. If a benchmark is not accurate, an enterprise can overload quickly causing customer dissatisfaction and profit loss

Data Points

Data points are the result of a benchmarking. They represent the summary of the specified results for a user load. Data points provide the detailed information required to make accurate decisions on the performance capability of a system-under-test. An example of a data point is transactions per second (TPS). Data points must be defined before implementing a benchmark.

Execution Plan

All testing processes require a well-designed execution plan. Execution plans ensure real-world environments duplication during a benchmark.

Control Logic

Control logic must have a well-defined initialization sequence that is separate from test and data collection. The control logic is not part of a benchmark. A benchmark depends on control logic to achieve:

Repeatability

Repeatability represents benchmark results from multiple runs of the same test with consistent configuration test environments and parameters. If a benchmark is not sufficiently repeatable, the results have no meaning.

Accurate Statistics

Accurate statistics are essential during a benchmark. Test result data, for example TPS, provides a user with the data necessary to determine if bottlenecks or potential enterprise problems may occur.

Implementation

Implementation of a benchmark specification can vary, depending on the system-under-test. The benchmark specification should always have audit trails to verify the implementation is performed in the “spirit” of a benchmark.

Benchmark Factory ® simplifies benchmark implementation. Benchmark Factory is a load testing and capacity planning tool for critical e-business environments. It places enormous stress on a system that is difficult to achieve in a standard testing environment. When a system breaks, it usually is under extreme load, and when it is needed the most. Identifying system capacity and per-

formance bottlenecks before they occur, Benchmark Factory facilitates proactive testing, and consequently reduces downtime, development costs, and potential loss of revenue.

Benchmark Factory allows a user to:

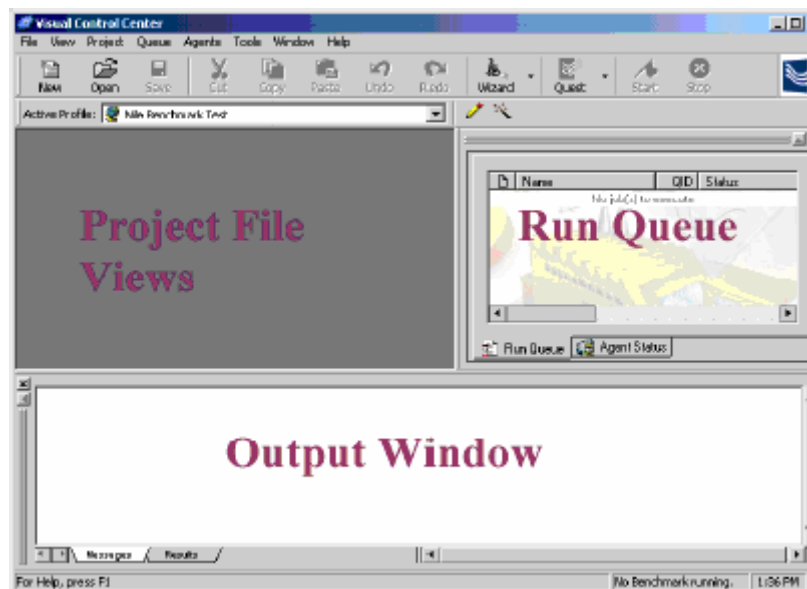
- Determine system throughput and capacity for Web, database, messaging, and file servers
- Create Web tests using a standard Web browser, spidering, or actual Web traffic history
- Examine test results with built-in analysis features and Excel reports
- Simulate thousands of concurrent users using minimum hardware

Benchmark Factory provides a Visual Control Center (Figure 1) as the graphic user interface to implement benchmarks. The Visual Control Center is comprised of the following areas:

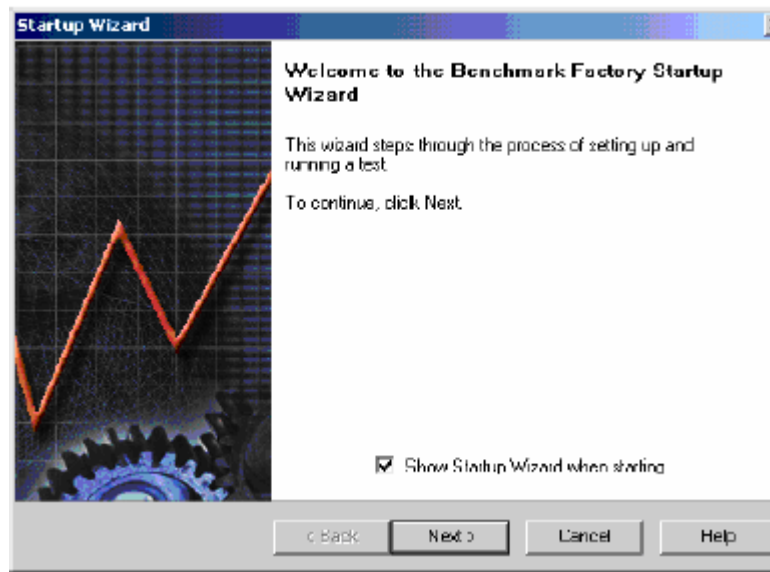
Project File Views—Displays test scripts associated with a project. A benchmark can contain several scripts, load scenarios, and user scenarios

Run Queue—A queue that controls the running of a job

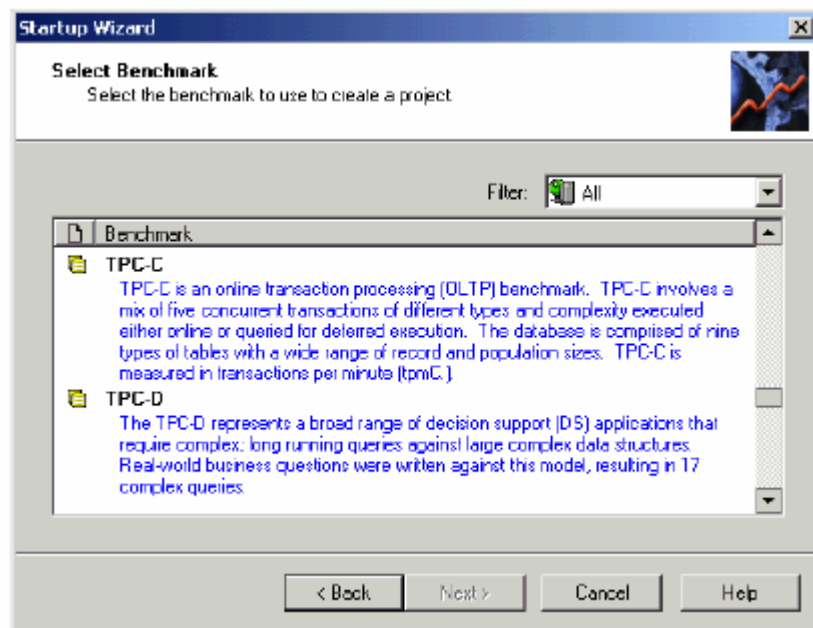
Output Window—Displays test data, and messages



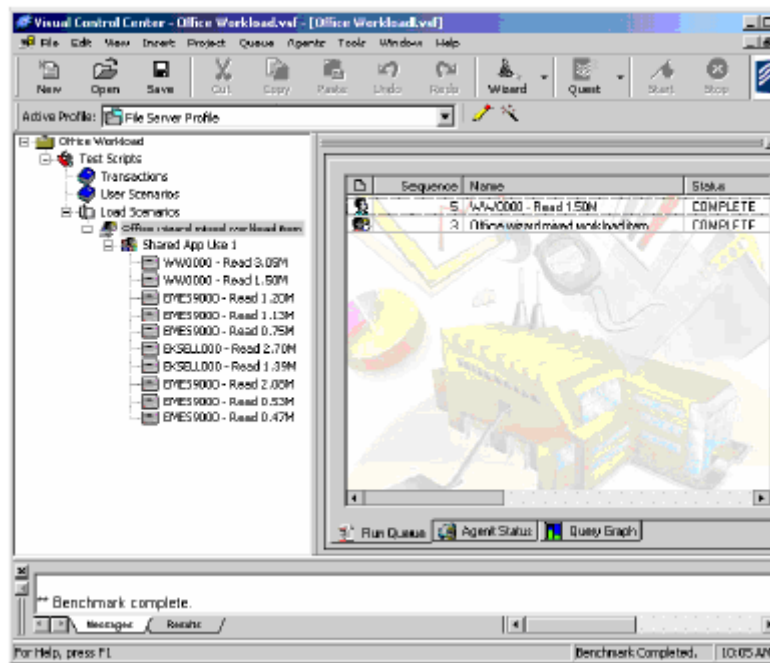
Benchmark Factory provides a Startup wizard (Figure 2) to streamline a benchmark testing process. The Benchmark Factory Startup wizard is a utility that creates mixed workloads for a benchmark. This wizard automates the testing process and assigns file extensions.



The Benchmark Factory Startup wizard provides the Select Benchmark feature (Figure 3). This feature selects a benchmark to use when creating a project.



After gathering the required information, the Startup wizard performs the necessary steps to create and run a desired benchmark successfully. Upon completion of a job, the Visual Control Center (Figure 4) displays and the results of a job can be viewed.



2.4. Conclusion

Basic benchmark components include specifications, control logic, and implementation. Benchmarking specifications consist of details, design goals, data points, and execution plans. Benchmarking specifications provide the documentation necessary and a roadmap to following during a test. Control logic provides repeatability and accurate statistics for a well-defined testing initialization sequence. Implementation of benchmarks varies depending on the system-under-test. Benchmark specifications should always have audit trails to verify that testing implementation is performed properly.

Benchmark Factory simplifies benchmark implementation. An effective e-commerce site must scale with the number of users, be reliable and quick, and optimize its use of bandwidth. Benchmark Factory provides the essential functions necessary to test e-Commerce sites.

In addition to Benchmark Factory, Quest Software offers a number of monitoring and tuning tools for e-commerce systems designed for capacity planning, load-testing process, and normal operation. These tools include:

Spotlight™—Provides essential functions to tune specific systems. Quest Software, Inc. offers Spotlight for Oracle and SQL servers

I/Watch™ —Provides essential functions to monitor and tune Oracle and SQL Server databases with trend analysis and drill-down capabilities

Foglight®—Provides complete application monitoring functionality