



# Automated Testing of Oracle Applications Release 11i (v.11.5.7)

A Test Automation Primer for Oracle Apps

White Paper



Jeffrey Spall  
Dan Downing  
Mentora Group  
Atlanta • Boston • DC  
[www.mentora.com](http://www.mentora.com)

## **Abstract**

This paper articulates the business context, challenges, and benefits of implementing a disciplined, automated approach for testing Oracle Applications Release 11i upgrades, using the Mercury tools WinRunner and TestDirector.

## **About the Authors**

**Jeffrey Spall** is a senior consultant with 14 years of IT Industry experience and 5 years testing the Oracle Applications suite using WinRunner, Load Runner, and TestDirector. He worked at Oracle as an Applications Consultant for two years, where he developed WinRunner scripts to test Oracle Applications.

Jeff has worked with customers such as Alcoa, Beckman Coulter, Caltech, Eaton, Qualcomm, and is currently working with Mentora Group at global manufacturer featured herein.

**Dan Downing** is the Vice President of Testing Services at Mentora Group. He is a 25-year veteran of the technology and software industry, and has held several executive management positions, and built two consulting practices on application performance and functionality testing.

Dan is the author of the *5-Steps of Web Stress Testing*, and has taught stress testing methodology and led load testing projects for the past 5 years. He is a regular speaker at Mercury and STAR conferences.

## **Table of Contents**

Abstract .....	2
About the Authors .....	2
<b>Why Companies Choose Oracle Applications .....</b>	<b>4</b>
<b>The Quality State of Oracle Applications .....</b>	<b>5</b>
<b>Benefits of Automated Functional Testing .....</b>	<b>7</b>
<b>The OA11i Test Environment .....</b>	<b>9</b>
<b>The OA11i Test Kit .....</b>	<b>10</b>
The OA11i Test Script .....	11
The OA11i Test Data Sheet .....	12
The OA11i Test Set.....	13
The OA11i Test Execution Report.....	14
<b>Case Study: Global Materials Manufacturer.....</b>	<b>15</b>
Project Team and Timeline .....	15
Ensuring Good Test Coverage .....	15
Expected Results and Benefits .....	16
<b>Summary.....</b>	<b>16</b>
About Mentora Group .....	17
About Mercury Interactive.....	17

## Why Companies Choose Oracle Applications

Organizations are looking for every available competitive advantage in today's business environment. To this end, IT executives look at their systems from two perspectives: gaining an information edge to improve real-time organizational responsiveness, and lowering operational costs.

In a survey conducted by InformationWeek for a September '02 article *Real-Time Business*, they report that

*“...3/4ths of the business-technology managers responding say senior executives ...have asked their IT departments to shorten the time it takes to deliver key operational data”*,

believing that their companies would gain a competitive edge by doing so.

James Hall, Managing Partner for Technology Services at Accenture, in a presentation at an Oct'02 *Executive Summit* ([http://www-svca.mercuryinteractive.com/cxo\\_corner/index.html](http://www-svca.mercuryinteractive.com/cxo_corner/index.html)) sponsored by Mercury Interactive, identified as the top two CIO Challenges “*Manage Costs*” and “*Integrate the Business*”.

According to the analysts, ERP package vendors are again showing strong growth, due to companies looking both toward optimizing their business and lowering the costs of maintaining legacy operational systems

Among the several reasons companies choose Oracle Applications (OA) are these:

- To lower the costs of maintaining legacy applications, by replacing them with an integrated suite
- For the rich functionality in Oracle's Financials and Manufacturing suites -- especially in the area of Project Accounting and Process Manufacturing -- which fit well with their organizational needs
- To leverage the in-house expertise they have developed by their use of the Oracle database to support other applications

Not surprisingly, they find that Oracle Apps are more challenging to implement than originally planned; no more problematic in total, perhaps, than other enterprise systems of the same scope, but problematic nonetheless.

Two significant challenges they encounter: *product quality* and *upgrading releases* once implemented.

## **The Quality State of Oracle Applications**

The Fall 2002 issue of the independent Oracle Applications User Group magazine, OAUG Forum (<http://www.oaug.com/>), an article on testing, cautions that

*“...the best insurance policy to protect the months and millions spent on careful planning and implementation is a disciplined testing methodology prior to go-live.”*

Underscoring this recommendation, among the 60+ attendees of the fall 2002 Atlanta OAUG (<http://www.atloaug.org/>) meeting, there was much talk about the need to validate quality through testing.

For example, in an article by McKee Foods in their Fall newsletter entitled The 10 Things We Learned During our 11i Upgrade, Lesson # 6 states:

*“Test everything! Don’t take anything for granted...it may be the one thing that will cause a delay to your go-live plans.”*

Then, in a Computerworld article on the OAUG Spring 2002 conference in Toronto, Boston-based AMR Research Inc. unveiled a report on users' reaction to Oracle 11i. While one of the findings was that 60% of OAUG member companies are either "satisfied" or "very satisfied" with Oracle as a company, another reveals that customer satisfaction with the 11i upgrade and implementation processes remains low.

Although companies reported business benefits from 11i, the study found that companies claimed they spent 43% more than they expected on an 11i implementation, and said it required 39% more time than they anticipated

AMR’s conclusion about these cost and time overruns: *"The root cause is quality."*

Finally, a subsequent InformationWeek (5/02) article entitled *Customers Clamor for Software Quality* (<http://www.informationweek.com/story/IWK20020512S0017>), Oracle acknowledged that the quality of their Oracle Application suite was negatively affecting their reputation with customers.

In that article, Oracle announced a broad action to overcome software-quality problems:

- training 1,000 developers to use Mercury’s WinRunner and LoadRunner automated software testing tools
- making every developer a quality-assurance tester
- implementing the company's first-ever formal development methodology with automated compliance checking
- releasing a customer Test Starter Kit Release 1.0 CD containing WinRunner scripts, Test Director Test Plan and Test Execution plans to ship with Oracle Apps 11.5.7

This paper proposes a testing solution for OA11i, and builds on the idea of a Test Starter Kit. Read on, and learn how to build an automated test environment to deliver high quality Oracle 11i implementations that reduce the high cost of manual testing following an initial investment.

## **Challenges of Upgrading Oracle Applications 11i**

Once Oracle Applications have been implemented, companies are faced with continuously upgrading configurations to alleviate the multiple unforeseen issues, as well as applying the stream of patches and *family packs* released by Oracle in response to problems surfaced by customers.

Among the multiple business and technical challenges associated with upgrading to or within OA11i:

- Refining business workflow changes imposed by new OA functionality
- Adding *flexfields* and customizing windows to accommodate enterprise-specific business needs
- Developing custom reports to support operational needs
- Designing and implementing a 3-tier web infrastructure with the right capacity to support the organization
- Coordinating multiple parallel efforts that must come together at various points to meet schedule milestones
- Conducting system acceptance testing with users prior to rollout to ensure that business processes can be executed successfully
- Managing the implementation to bring it in on schedule and on budget.

These adjustments -- no matter what form and no matter the reason they are implemented -- force the organization to continuously retest. Since many of the companies implementing OA11i are not in the application development business, and thus do not have seasoned software quality assurance departments, this effort relies on enrolling business users to perform the testing which must be repeated time and again for each patch implementation.

The hidden cost that companies uncover quickly is that business users are difficult to tap continually for testing, and yet at the same time they uniquely hold critical knowledge about the use of the system necessary for good testing.

As a result, the solution that IT managers are turning to is *automating* this testing using tools that enable codification of the business user's knowledge into repeatable test scripts.

## **Benefits of Automated Functional Testing**

The objective of automated testing is to establish a regression test kit that can be updated with each new release, and that can be used repeatedly without having to tap the business users time and again.

Codifying business user's knowledge to automate future testing begins with establish solid testing requirements that ensure *sufficient test coverage* of the OA modules that are implemented.

Test *coverage* – the number of business functions that are tested – grows over time as more manual testing scripts are automated. Estimates obtained from surveys reveal that IT teams utilizing automated testing tools gain somewhere between 200 to 300 percent of test coverage per Oracle R11i module, compared to the time and resource-intensive manual testing.

Achieving good test coverage takes thoughtful planning, continued development and maintenance of test libraries, and an understanding of the key business processes in each OA module.

Once the business user's knowledge has been codified into a repeatable test script, and these scripts have been logically sequenced to perform a complete business function, IT can re-run these scripts with a marginal cost that is a fraction of the cost of manual testing.

The regression testing process, including a complete regression test kit for OA and the review, investigation and response to any deviations that are identified can still take from days to weeks to complete. Therefore, a quality system framework is required to assure that time and methodology is planned into the organization's change control process.

The benefits of automating functional testing thus are:

- Greatly reduced cost of testing, over time, following the initial investment to develop the regression testing kit
- Higher quality production releases, because of:
  - the ability to run tests more often
  - in a shorter time window
  - with less chance of human error
  - with regression coverage of modules that are not directly affected by the patch or family pack
  - with incremental coverage increases with each patch or family pack as more manual testing scripts are codified.

Additional benefits of automated testing include:

### **Quality Improvements**

- Reduced number of production defects. While testing does not *reduce* the number of defects, it *does* reduce the likelihood that they will migrate into production.

- Reduced testing hours. The number of resource hours is reduced significantly; the testing duration is shortened less dramatically, as test suites still have to be run and be analyzed.
- Reduced number of help-desk calls. Anomalies and organizational work-arounds are defined ahead of time, as a result of testing
- Reduced risk of lost business due to software problems

**Responsiveness to Customers and End Users**

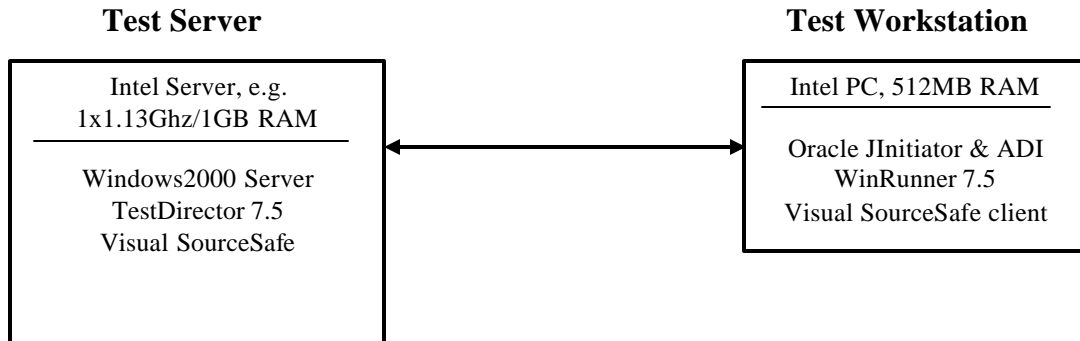
- An “educated” decision that the new release/upgrade/patch is good enough to be rolled into production
- Increased customer satisfaction with the new release
- Upgraded quality disciplines within IT that in turn deliver value to the organization

## The OA11i Test Environment

To achieve these benefits, IT needs to invest in developing and maintaining an OA11i Test environment, consisting of a computing platform that houses an evolving OA11i Test Kit.

The hardware configuration can vary, but often consists of a dedicated server that houses the TestDirector repository and a robustly-configured testing workstation on which WinRunner is installed, for each of your testers.

**Diagram 1: Typical Test Environment**



The roles of the automated testing tools from Mercury are as follows:

- WinRunner – This is the automated tester’s workhorse, which enables:
  - Keystroke recording to capture the business user’s session
  - Linking Excel spreadsheets to user-entered data to create repeatable and iterative data-driven tests
  - Customizing scripts to the target application with script steps to, for instance, confirm correct program execution, iterate multiple times, conditionally branch if an error messages is displayed, generate unique data key values where needed, etc.
  
- TestDirector – This is a web-accessible central repository which stores and manages the testing process by enabling:
  - Sequencing of individual scripts into complete business workflow tests. Once sequenced, as one script completes with the expected results, it triggers another to begin until the workflow is complete.
  - Scheduling hands-off execution of script sequences
  - Storing in organized directories all testing work products: test requirements, scripts, results
  - Logging, tracking, and reporting defects and their resolution status.

## The OA11i Test Kit

The Test Starter Kit released by Oracle as part of OAi-11.5.7 provides test scripts designed by Oracle's own QA departments. It provides a dozen or so scripts per application group as a starting point for customers that want to automate their testing.

We have expanded on this Test Kit to create a robust repository of reusable WinRunner scripts, catalogued and linked within TestDirector, to provide good coverage of Financials and Manufacturing processes.

The key components of our OA11i Test Kit are:

- **Test Scripts:** A collection of WinRunner-recordings of the human interaction with the user interface (keystrokes and mouse clicks) that represent a specific function within a specific OA11i module, e.g., *Create a Sales Order* in the Order Management module.
- **Test Data Sheet:** Sets of values that are inserted at execution time into user-entered fields. WinRunner automates the creation of these data sets and stores the data in easily readable and maintainable Excel spreadsheets
- **Test Sets:** Sets of sequenced scripts that execute a specific test objective, for instance, a complete business function that crosses multiple OA modules
- **Test Execution Reports:** Automatically generated results of test set runs. These reports provide summary information, such as execution time and status, name of tester, pass-fail on the overall test set, and then provide drill-down into the detailed steps of the run.
- **Data Templates:** These are optional Excel spreadsheets that can be used by data-loading tools, such as Oracle's *SQL\*Plus* or *Tools for Oracle Applications Developers* (TOAD) for uploading data into the test Oracle database, to ensure that data expected by a test set exists.

Once developed, an OA11i Test Kit has multiple uses:

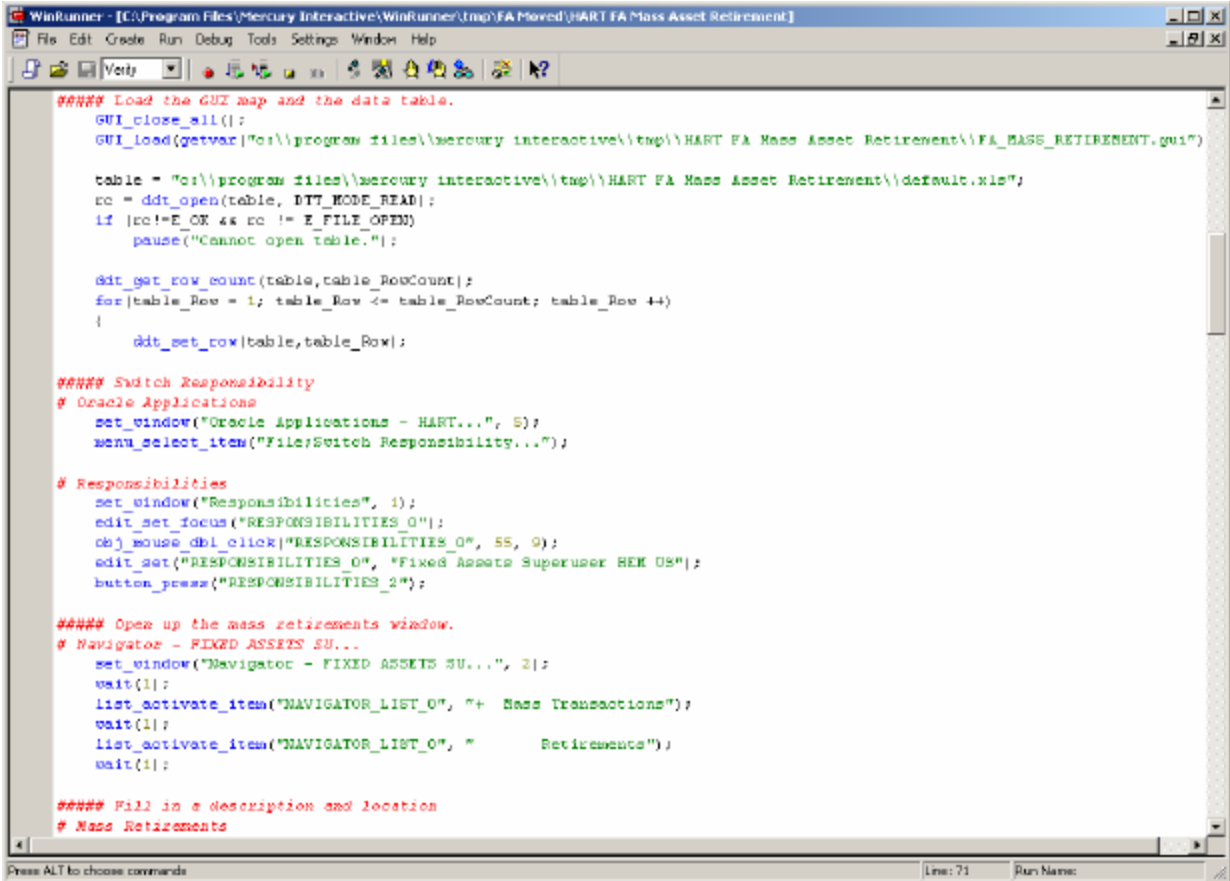
- Automation of test data set-ups, using TOAD and Data Templates
- Individual module testing, using individual or groups of test scripts to test customized OA11i screens and reports
- Integrated system testing, to test complete business workflows
- Regression testing, to test the integrity of the base functionality of a patch or a new release.

These components are discussed in detail below.

## The OA11i Test Script

The Test Script is the starting point and fundamental building block of an OA11i Test Kit.

WinRunner automates the creation of test scripts by recording a business user's navigation through a business process. The result is a readable and customizable script that uses Mercury's function-rich *Test Script Language* (TSL) API. It shows the steps of the workflow navigation, user-entered data, and any system-generated data returned by the application.



```
##### Load the GUI map and the data table.
GUI_close_all();
GUI_Load(getvar|"c:\program files\mercury interactive\tmp\HART FA Mass Asset Retirement\FA_BASE_RETIREMENT.gui")

table = "c:\program files\mercury interactive\tmp\HART FA Mass Asset Retirement\default.xls";
rc = ddt_open(table, DDT_MODE_READ);
if (rc!=E_OK && rc != E_FILE_OPEN)
    pause("Cannot open Table.");

dtt_get_row_count(table,table_RowCount);
for(table_Row = 1; table_Row <= table_RowCount; table_Row ++ )
{
    dtt_get_row(table,table_Row);

##### Switch Responsibility
# Oracle Applications
set_window("Oracle Applications - HART...", 5);
menu_select_item("File;Switch Responsibility...");

# Responsibilities
set_window("Responsibilities", 1);
edit_set_focus("RESPONSIBILITIES_0");
obj_mouse_dbt_click|"RESPONSIBILITIES_0", 55, 9);
edit_set|"RESPONSIBILITIES_0", "Fixed Assets Superuser BEN US");
button_press|"RESPONSIBILITIES_2");

##### Open up the mass retirements window.
# Navigator - FIXED ASSETS SU...
set_window("Navigator - FIXED ASSETS SU...", 2);
wait(1);
list_activate_item("NAVIGATOR_LIST_0", "+ Mass Transactions");
wait(1);
list_activate_item("NAVIGATOR_LIST_0", "      Retirements");
wait(1);

##### Fill in a description and location
# Mass Retirements
```

Example of a test script shown in the WinRunner interface

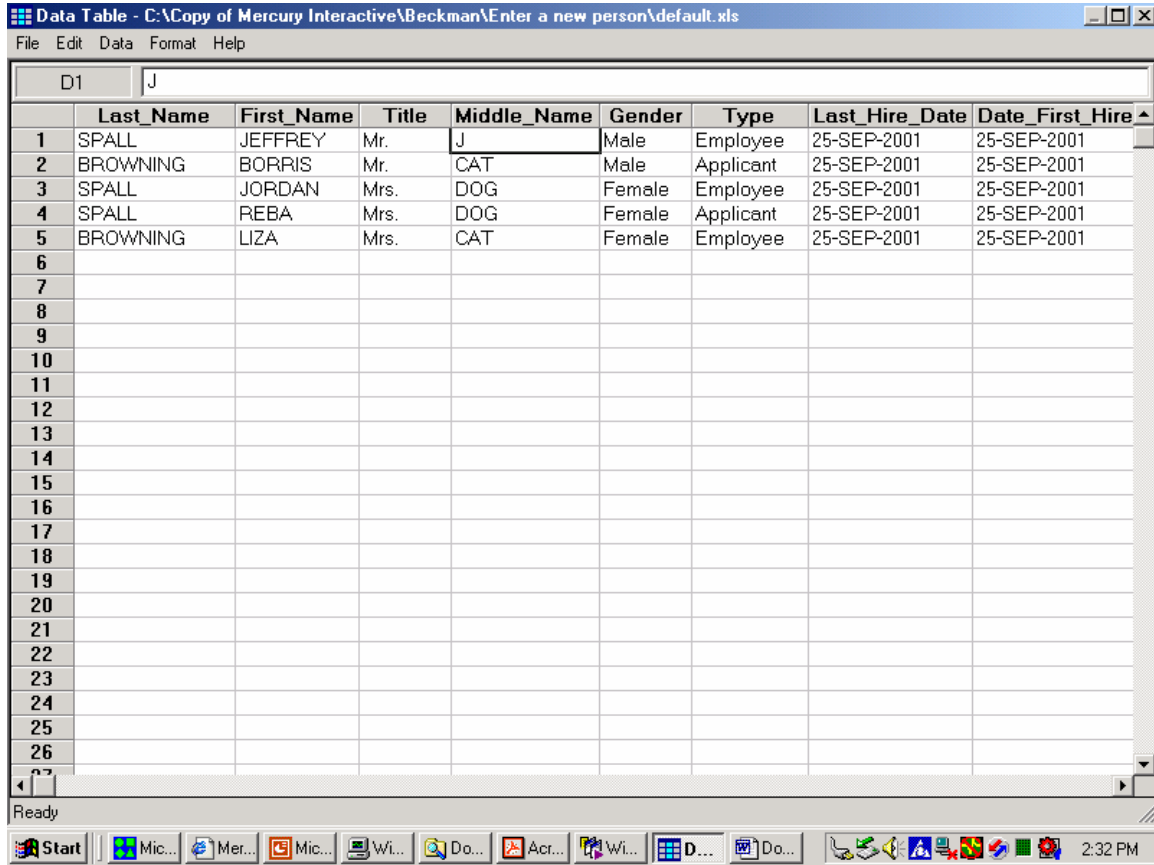
Once recorded, you use WinRunner's facilities to customize and enhance the script by adding data sets, iteration loops, conditional execution, transaction markers for timing key operations, and more.

The skill-set required to master WinRunner builds on good software quality and analytical disciplines, and adds the need to be comfortable learning a scripting language. Some previous programming or scripting experience is very helpful.

## The OA11i Test Data Sheet

An OA11i Test Data Sheet is a Microsoft Excel spreadsheet consisting of data variables that have been identified as user-entered data in the test script. One sheet is created for each set of data values and is linked to the script such that, at execution time, values from succeeding rows are entered into the test at each iteration.

These data sheets enable data-driven testing, where known inputs can be checked against expected results to validate that a function has executed correctly.



The screenshot shows an Excel spreadsheet titled "Data Table - C:\Copy of Mercury Interactive\Beckman\Enter a new person\default.xls". The spreadsheet has a grid with columns labeled "Last Name", "First Name", "Title", "Middle Name", "Gender", "Type", "Last Hire Date", and "Date First Hire". The rows are numbered 1 through 26. The data in the first five rows is as follows:

	Last Name	First Name	Title	Middle Name	Gender	Type	Last Hire Date	Date First Hire
1	SPALL	JEFFREY	Mr.	J	Male	Employee	25-SEP-2001	25-SEP-2001
2	BROWNING	BORRIS	Mr.	CAT	Male	Applicant	25-SEP-2001	25-SEP-2001
3	SPALL	JORDAN	Mrs.	DOG	Female	Employee	25-SEP-2001	25-SEP-2001
4	SPALL	REBA	Mrs.	DOG	Female	Applicant	25-SEP-2001	25-SEP-2001
5	BROWNING	LIZA	Mrs.	CAT	Female	Employee	25-SEP-2001	25-SEP-2001
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								

**Example of a test data sheet shown in the Excel interface**

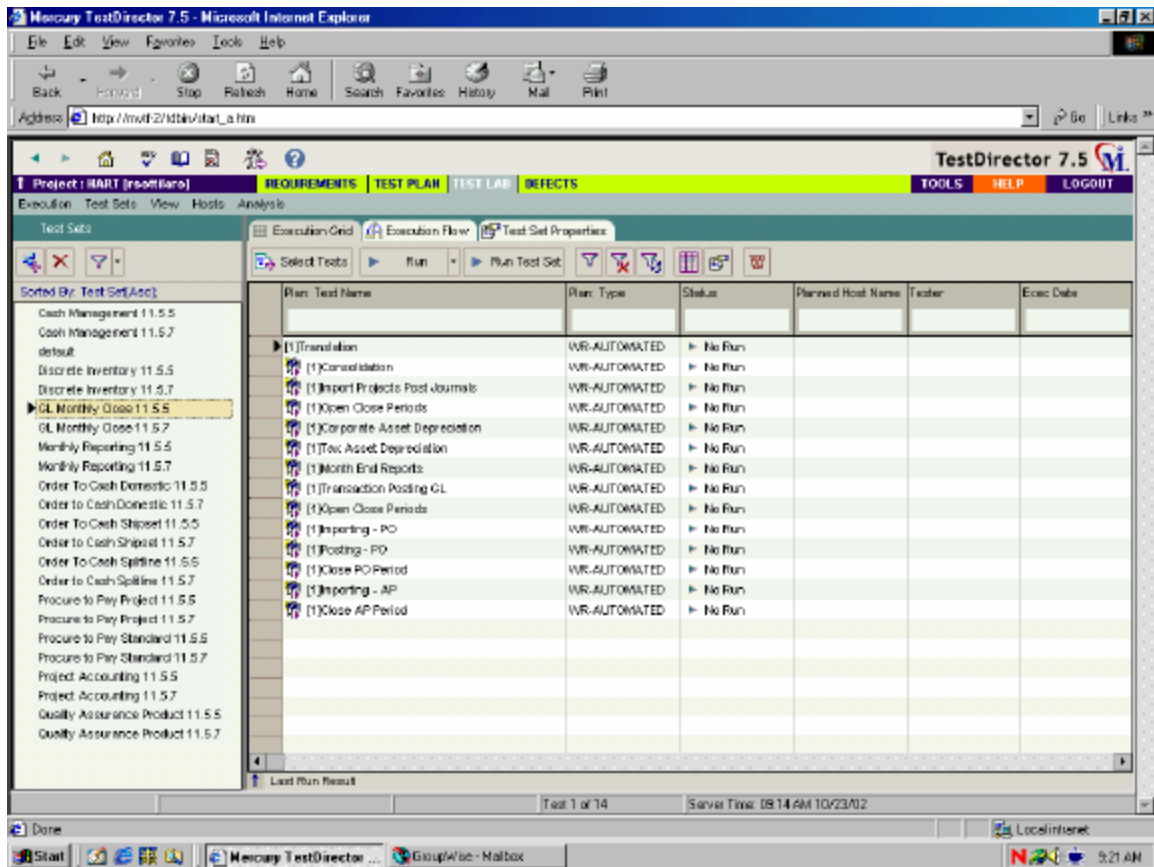
The WinRunner-guided process to link user-entered data with values in a data sheet is called *parameterizing*. This process has the following steps:

1. Walk through the steps of using an application function, letting WinRunner record and generate the script
2. Interactively use the WinRunner parameterization facility to find and replace user-entered values with variable tokens, and link these to an Excel sheet where additional values are entered
3. Manually add a loop statement to the script to iterate a number of times, each time selecting data from a different row of the data sheet.

## The OA11i Test Set

An OA11i Test Set is a group of test scripts related by business function, application module, or test objective. You define a test set within TestDirector, by selecting test scripts from the test plan tree, and assigning this group of tests a descriptive name.

For example, the individual scripts used to create a supplier, create a purchase order, issue a requisition, receive the items, and post to the General Ledger, are linked together so that successful completion of one script launches the next, testing the complete function.



Example of a test set in TestDirector

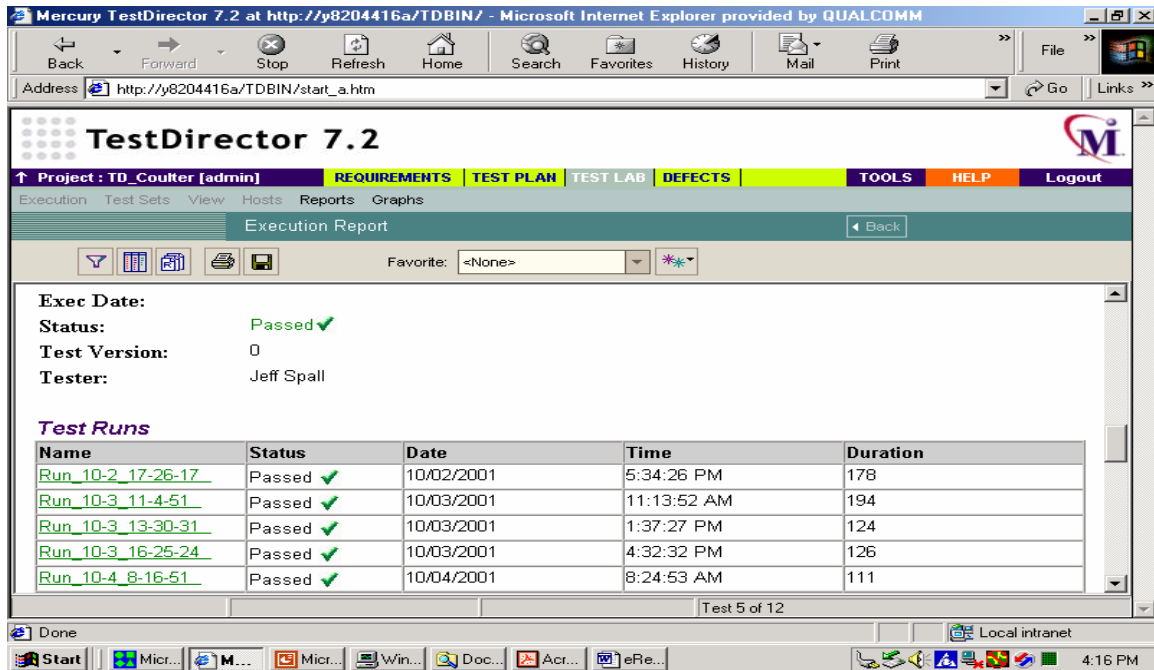
The figure above shows the Test Set, a sequenced set of scripts in TestDirector.

TestDirector also enables:

- Documenting of test requirements
- Scheduling hands-off test runs
- Tracking test results against test requirements
- Web-enabled defect tracking to log, route, and status defects
- Interfacing with source code control systems to maintain the integrity of test scripts.
- Tracking and reporting test results, including defects per patch/development cycle

## The OA11i Test Execution Report

Once an **OA11i** Test Set is scheduled and executed, you can view within TestDirector the test results in a Test Execution Report.



Example of a test report shown in the TestDirector interface

The report-writer embedded in TestDirector can also be used to generate summary tabular and graphical charts for project meetings to present the quality progress of a release.

## **Case Study: Global Materials Manufacturer**

One of Mentora's customers is a global materials mining and manufacturing company with plants throughout the US and northern Europe.

They are 18 months into an Oracle Applications 11i implementation that includes Financials and Discrete and Process Manufacturing modules that are replacing a dozen legacy applications, and will ultimately be rolled out to all their manufacturing facilities.

The organization has been live on OA11i for about 9 months. They have relied on a handful of business analysts and on approximately two dozen users to perform manual testing before each major go-live event. In so doing, they have experienced numerous patches from Oracle, the need for careful testing prior to implementing patches, and the high cost of having to rely on the scarce business resources for testing.

These expenditures of internal capital have forced them to consider alternatives as they planned a migration from 11.5.5 to 11.5.7 and a 'wave 5' rollout to their European plants. An approach that could automate this testing made good business sense, and its implementation is in process, in partnership with Mentora.

### **Project Team and Timeline**

Creating this regression test repository and testing the 11.5.5 and 11.5.7 releases prior to converting the production application, required a team of five people for about a fourteen week project period – a sizable initial investment, but one that is expected to pay dividends going forward.

The Mentora team consists of a half-time project manager, a senior OA11i Expert, and three WinRunner developers. The supporting customer's team that provided key application workflow knowledge consists of a dozen business analysts and business users, a part-time project manager, and a strong IT executive project sponsor.

### **Ensuring Good Test Coverage**

As discussed above, one of the keys of a good test plan is to test sufficient processes and thus ensure sufficient test coverage to provide confidence in the overall quality of the release. To ensure this, a careful Discovery process to inventory and prioritize key business functions was undertaken, guided by a senior quality consultant, and involving many of the customer's business analysts and users.

We found that adequate coverage resulted in somewhere between 15 and 25 scripts per module, and a total of just over 200 scripts for the thirteen deployed modules in Financials and Manufacturing.

A good disciplined methodology for documenting the test plan and the detailed methodology, tracking issues, and for executing linked test scripts is also a key success criterion for this project.

## **Expected Results and Benefits**

The expected results of this investment align with those stated above in the case for automates testing: higher quality releases, and reduced testing costs.

The OA Test Kit will be turned over to the customer, with specific training for a couple of people to enable them to carry on the effort of maintaining existing scripts, creating new ones, and executing tests as often as patch-upgrade needs dictate.

The customer's *Quality IQ* will have been raised several notches, and IT's overall standing with their business customers should be enhanced as well.

## **Summary**

The quest for quality Oracle R11i Applications is driven by the high costs associated with manual testing and the large number and high frequency of software patches released by the vendor.

Automated testing is an IT insurance policy that provides protection against costly business interruption risks and yields a solid ROI over the lifecycle of an implementation, when compared to manual testing, and an even higher ROI when compared to de-emphasizing or skipping regression testing.

Following a significant initial investment, automated tools reduce the test cycle time for testing and enable the organization to test often at a low marginal cost.

Automated testing tools provide a platform around which organizations can implement a disciplined testing methodology, to ensure high quality in their production applications.

The savings in productivity, re-work, maintenance, and business reputation are all benefits that accelerate over time.

## **About Mentora Group**

Mentora Group, Inc. is a technology services company, founded in 1994, with offices in Atlanta, Boston, and Washington DC.

The company's byline is *Dependability for Business-Critical Deployments*, and its services bridge the gap between application development and production management, as well as providing application-level managed services through their partnership with AT&T Internet Hosting data centers.

They provide functional and performance testing services for Oracle Applications, PeopleSoft, and other enterprise web and client server applications.

Mentora is a Mercury Interactive Channel Alliance Partner.

For more information about Mentora: [www.mentora.com](http://www.mentora.com)

## **About Mercury Interactive**

Mercury Interactive, the global leader in business technology optimization (BTO), delivers Optane(tm), a suite of integrated products for enterprise testing, production tuning and performance management, that enables customers to optimize business processes and maximize business results. Customers worldwide - including 75% of the Fortune 500 - use Mercury Interactive solutions across their application and technology infrastructures to continuously measure, maximize and manage performance at every level of the business process and each stage of the application lifecycle to improve quality, reduce costs, and align IT with business goals.

Founded in 1989, Mercury Interactive is headquartered in Sunnyvale, California, with offices in more than 25 countries. Further information is available at [www.mercuryinteractive.com](http://www.mercuryinteractive.com) <<http://www.mercuryinteractive.com>> or by phone at U.S. +1.408.822.5200. The company's common stock trades on the Nasdaq National Market under the symbol MERQ.

## **About Oracle**

Oracle Corporation is the world's largest enterprise software company, providing enterprise software to the world's largest and most successful businesses. With annual revenues of more than \$9.4 billion, the company offers its database, tools and application products, along with related consulting, education, and support services.

Headquartered in Redwood Shores, California, Oracle is the first software company to develop and deploy 100 percent Internet-enabled enterprise software across its entire product line: database, server, enterprise business applications, and application development, and decision support tools.

For more information about Oracle: [www.oracle.com](http://www.oracle.com)