

# **Developing A Long-term Testing Strategy for Your Organization.**

Connie Cook  
*Liggett Group Inc.*

Bill Dunham  
*BOSS Corporation*

## **Introduction**

The Liggett Group, Inc. ("Liggett") develops testing strategy for the year 2000 and beyond. The objective of this testing strategy was to *minimize* and *expedite* the tasks related to test script development, execution, documentation, and overall management of an Oracle Application testing effort. Liggett has been an Oracle Applications customer since 1993 with the following modules installed: GL, FA, AP, PO, WIP, BOM, CST, INV, MRP, CM and a host of third party products.

## **Liggett Group Inc. Overview**

Liggett is the fifth largest manufacturer of cigarettes in the United States in terms of unit sales. Liggett is the operating successor to Liggett & Myers Tobacco Company, which was formed in 1873. Liggett is an indirect subsidiary of Brooke Group Ltd., a company listed on the New York Stock Exchange.

## **Executive Summary**

The objective of this testing was to minimize and expedite Oracle Application testing through the development of a repeatable Testing Strategy for use during a year 2000 project and future upgrades and enhancements. Coming fresh off an upgrade to 10.7SC 16.1, which required the development and execution of testing documentation, and having no available soft copies to utilize, Liggett was determined to develop a testing strategy that would support them for their year 2000 testing and beyond. With an understanding that the future releases of Oracle Applications would change, the test scenarios and scripts were developed generically enough to work for Liggett's current release, and for Release 11 or Release 11i upgrades, therefore leveraging their investment.

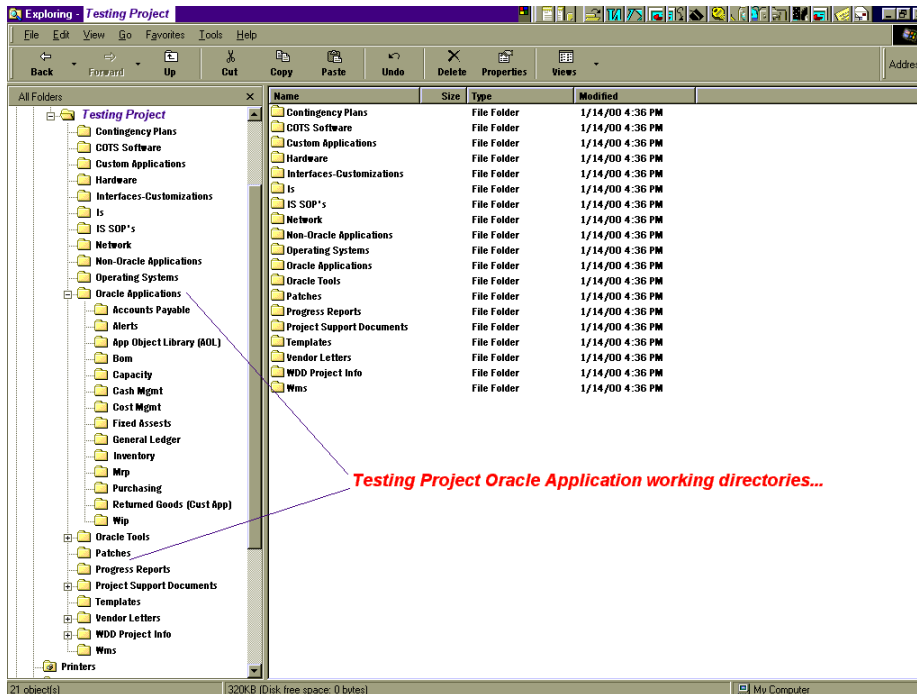
## **Team Selection**

The first step taken was to assemble a cross-functional team. The team was made up of Information Services Department personnel, functional members (each supporting the various Oracle Applications and third party products), a Liggett project sponsor, and a consulting firm to manage the project and provide functional and technical assistance as needed. Once the team was selected the planning phase began.

## **Testing Strategy**

To kickoff the project an initial planning session was held with all team members participating. All team members were assigned specific roles on the project. The objective was to have at least one person from each area of the business participate in the testing. Each person would be assigned a specific group of Oracle Application modules, third party products, or customizations to test.

As part of the testing strategy, a project-working directory for all project documents was created. This repository was used to store the testing strategy, application tests, output files, vendor letters, TAR and patch information and project progress reports. The directory structure was broken down so that each Oracle Application had its own working sub-directory. Within each directory, the application testing scripts were stored for easy retrieval and execution for each team member.



## Assignments: Testing Scenarios & Scripts

The testing strategy was first used to validate Oracle Corporation's statement that Oracle Applications were Y2K compliant. All testing scenarios and scripts were developed generically enough that they could be easily changed to support future application upgrades to Release 11 or Release 11i.

As the project continued, assignments were made to each functional team member to identify specific business scenarios that required thorough application testing, specifically where dates were concerned. A list of 119 tests were identified consisting of the following areas; Oracle Application modules and customizations, hardware, operating system, non-Oracle Applications, network, custom applications, COTS software and IS standards. Out of 119 tests created, 49 were actually executed. Vendor compliant letters or internally accepted documentation supported the remaining 70 tests. The 49 tests that were created related directly to the primary Liggett business functions. These tests supported the installed Oracle Financial and Manufacturing Applications, as well as customizations and interfaces.

Each test was broken down into detailed test script. These scripts were then further broken down into specific action steps that provided additional accuracy and verification of the business scenario. There were over 1400 test steps executed during the testing effort. During the year 2000 testing effort these tests were executed four times for varying system dates to ensure the application was properly tested.

To help document the tests, each team member was given soft copies of templates for use in developing test scenarios and subsequent testing scripts. These testing scenarios and scripts will be retained for future application testing as well as a basis for future end user training material. As new releases of the Oracle Applications become available and Liggett goes through subsequent upgrades to Release 11 and/or Release 11i, these documents and the project teams time and effort will be leveraged.

**Oracle General Ledger**

Scenario #	Test Name	Description	Date	Time Estimate to Complete (Hours)	Tester (Liggett)	Results (Passed, Failed)	Accepted by (Liggett)
1	Open Period	Open new period for Y2K	10/18/99	.25	C Cook		
2	Close Period	Close prior period for Y2K	10/18/99	.50	C Cook		

Total Estimated Hours to Complete Tests: .75

The above template example lays out a sample of the testing scenarios for Oracle General Ledger (this is a very simplistic example, primarily used to show the template format, not content). These scenarios are used to capture the specific areas of the business and Oracle Applications to be tested. The document is considered to be the first step in creating end user testing documentation.

From left to right the template is as follows. The Scenario number is a sequence identifier for each test that will be executed. Test Name is the name given to the test by the person it was created by, and a Description is a simple explanation of the test. The Date column was used to determine the date tests would be executed and for the scheduling of testing resources.

The Time Estimate to Complete column was used to capture the time required to execute the test scenario. The time was estimated by the end users that were most familiar with the testing scenario. You will see a total estimate for the scenario tests at the bottom of the table above. This estimate was transferred to the project plan in order to have an estimated completion for all testing tasks. This module test time was combined with all other test estimates, to determine the total time required to complete the project. The next column, Tester, is for the name of the person executing the Test. The final two columns, Results and Accepted By, are for monitoring the tests for Pass or Failure, as well as end user acceptance.

**Testing Steps for Scenario: Oracle General Ledger #1**

Please list below each step that needs to be executed for the above test.

Scenario #	Test Step	Role/Responsibility	Module	Action	Expected Results	Actual Results	Result (Passed, Failed) / Reason / Issue# / TAR# / Tracing #
1	1	GL Super User	GL	Open GL Period	Period Opens successfully		
	2						
	3						

The Testing script example above lays out the steps required for the Oracle General Ledger Scenario. This outlines the “step by step” approach to application testing and is used by the application testers to guide them through the testing scenarios previously identified.

From left to right the template is as follows. The Scenario # was used to link the testing scripts to the test scenario for Oracle General Ledger above. The Test Step was a number assigned to each unique step in the testing script. The Role and/or user Responsibly identified the resource and application access required to execute the test step. The Module Name, to be clear, is the Oracle Application module in which the test step is to be executed. (This could be different from the test scenario.) The Action column described what action to take when executing the test. An Action example could be the application navigation path or keystrokes required to execute the test. The Expected

*Results* column was used to provide information on what should happen after the *Action* was taken. The *Actual Result* column captured the actual testing results and was compared to the *Expected Results* column to ensure testing success. Testers determined whether the test passed or failed by comparing the two columns. If the test failed the tester would provide supporting comments as to why it failed and would add the associated Issue, TAR or tracking number to assist with the resolution process.

### Testing Work Breakdown Structure (Project Plan)

Once all the test scripts were created, and time estimates were provided, they were added as tasks to the testing project plan. This plan was used to manage and control the project throughout the duration. A typical project plan was created using MS Project, with phases, activities and tasks. The project plan consisted of a *Task ID number*, *task name*, *duration*, *start and end dates*, *percent complete*, *task predecessors* and *resource names*. The final plan had over 1400 tasks within activities such as Oracle Application Software testing to testing the interfaces between the Oracle Applications and Liggett's computerized maintenance management software (CMMS) product.

**Liggett Group Inc**  
**Oracle Applications Testing Project**  
**11/19/99**

ID #	TaskName	Duration	Start Date	Finish Date	% Complete	Predecessor	Resource Names
1.	Testing Project Plan	64 days	Fri 9/17/99	Fri 12/17/99	69%		Connie Cook
2.							David Channell
3.							Lisa Over

The project plan was distributed to all project team members and was used for monitoring assignments and project progress. The plan was saved into the Testing Project working directory and was made readily accessible to all project team members. The project plan was updated weekly and submitted to the project manager for review and approval.

### Capturing Test Issues

As testing progressed, logs were used to capture all test-related *issues*. Information such as the *Oracle Application module* and *Issue number* within the module where captured. Additional information such as the *description* of the issue, *who was assigned* to the issue for resolution, *when* the issue was expected to be resolved, was also captured. Information such as the *impact* to the organization, the specific *status*, and *assignment* of any TAR or tracking related information or comments were also tracked. All team members were required to log their own issues and forward them to the management team on a weekly basis.

Liggett Group, Inc.		Testing Project Issues					BOSS CORPORATION	
		As of 11/19/99					Member Organization Service Agreement	
App Module	Issue#	Description	Assigned To (Initials)	Priority (P/low)	Expected Resolution Time (Days)	Organization Contact	Current Status	Issue # / TAR # / Tracking # / Comments
INV 203750	-	Forecasted production and Expansion of Production has incorrect data - submitted by Bill	LC (initials)	H	2 wks	Client complete testing must be received	A	Have asked for code to be sent to Bill Dunham for delivery to technical person at BOSS to investigate the problem.

### Testing Status Reports

Each week *status reports* were developed which provided information such as the *testing project completion percentage*, overall *Liggett and BOSS team progress* for the week, the number of *Tests performed* for each testing

iteration and a consolidation of all *project issues* into one master issues log. Each status report was saved to the Testing Project working directory for future reference and submitted to the project manager each week.

**Liggett Group, Inc.**      **Testing Status Report**      **BOSS CORPORATION**  
*As of 11/19/99*      Better Organization Service Solutions

Overall Percent Complete: 33%  
 Test Percent Complete (Third Iteration): 73% (See MBS for details)

**1 Overall Test Progress: Week of 11/15/99**

**Liggett Team Accomplishments:**

*Monday, 11/15/99*

*Tuesday, 11/16/99*

*Wednesday, 11/17/99*

*Thursday, 11/18/99*

*Friday, 11/19/99*

**BOSS Team Accomplishments:**

**2 Quantity of Tests Performed: First Iteration**

Tester	Total	Tests Passed	Tests Failed (-)	Tests Outstanding
Connie Cook	37	37	0	0
Bonnie Lee	86	78	0	8
Shirley Jackson	0	0	0	0
Elke Hoffman	19	19	0	0
Cari Spray	52	51	0	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
<b>Totals</b>	<b>343</b>	<b>321</b>	<b>0</b>	<b>22</b>

**Liggett Group, Inc.**      **Testing Status Report**      **BOSS CORPORATION**  
*As of 11/19/99*      Better Organization Service Solutions

App Module	Issue#	Description	Assigned By (initials)	Priority (below)	Expected Resolution By (date)	Organization Impact	Current Status	Issue # / TAR# / Tracking # / Comments
INV @2960	1	Forecasted Production and Extension of Production has incorrect data - submitted by B Lee	L Overton	H	Asap	Can't complete testing run; to be resolved	A	Have asked for code to be sent to B Durham for delivery to technical person at BOSS to investigate the problem.

To help manage and control the testing project each test was monitored and whether it passed or failed, was documented in section 2 of the status report. If an issue such as an Oracle Application error caused a test to fail, or a test step didn't meet the expected results, it was determined failed and documented as an issue and placed in the issues log for immediate evaluation and resolution. The issues log was reviewed and updated each week. Issues that were closed or resolved were changed to italics and printed for one subsequent status report, then removed. The primary objective of the status report was to effectively communicate project progress to management, the project team and end users.

**Contingency Plans**

This section discusses the creation of *Contingency Plans*. You will find an example of a contingency plan template below. These plans were created to support Liggett's operations in case the testing scenarios didn't cover all the processes of the application and business. The plans were also used as backup procedures for all employees in case something went wrong during the cutover to production. A contingency plan was created for each area of the organization. These plans included the associated *Test ID Number*, the *Dependency Name* and potential *Impact* to the organization if the process failed. There was also a column that documented the *Preparation* required to support the contingency plan. This allowed users to proactively prepare for problems and issues versus reacting.

Risk Scale: High = 1, Medium = 3, Low = 5

Test Id Number/Name	Dependency Name	Impact	Preparation	Reaction	Transition	Resumption	Level of Inport	Prob. of Failure	Failure Impact	Risk Rating
1.1 Receive Inventory	Oracle Applications: Inventory module must be accessible	Need method of capturing materials received.	Develop master document for tracking all incoming materials	Maintain daily record of materials received and provide to Purchasing, Materials Planning	Add materials received to master document	Using master document, enter all receipts	1	5	1	2
1.2 Return Inventory	Oracle Applications: Inventory module must be accessible	Need method of capturing materials returned	Develop master document for tracking all returned materials	Maintain daily record of materials returned and provide to Purchasing, Materials Planning	Add materials returned to master document	Using master document, enter all returns	1	5	3	3

The *Reaction* column documented what the users were to do in case something went wrong. It allowed the users to react as quickly as possible to a failure with little or no confusion on what was to be done to resolve the issue. The *Transition* column was used to describe the procedures that were required to support the task during the period it was unavailable. The *Resumption* column was used to explain how to catch-up or resume normal processing after the problem was resolved. The final section helped document the *Level of Risk* to the organization if the process failed. The *Level of Importance* to the business was determined on a scale from 1 to 5, with level 1 representing highly important. The *Probability of Failure* was also captured, it was based on team members knowledge of the Oracle Applications and business processes. This documented the probability of a task failing when the system transitioned into production. The *Failure Impact* was a measure of severity to the organization if the process or task failed. This estimate was also provided by the team members, it documented their understanding of the impact to the organization if the process or task failed. The last column documented the overall calculation of *Risk*. If a contingency plan calculated a risk rating of 2 or below it was closely monitored during the transition period due to its high probability of failure. This was an easy and simplistic way to proactively prepare for our transition to the production application.

## Testing Summary

At the end of the project a *Testing Summary* document was created to identify project successes and failures. As part of this summary, Liggett's management wanted a document to make all testing related information easily accessible. To provide this request, hyperlinks were created throughout the document linking each test scenario and output, as well as the contingency plans to its supporting documentation with a click of a button. This fulfilled management's request and provided an excellent way to keep documentation on-line and readily available.

Oracle Applications – Master Test Owner: [Cornie Cook](#)

Item#	Test Description (what)	Responsibility (who)	Status (compliant, non-compliant)	Priority	Letter/Issue#/TAR/ Patch/ Tracking#	Comments
22	Oracle Apps Assets 10.7SC	R. Yates C. Spray	Compliant January 3, 2000 February 29, 2000 October 10, 2000 December 31 2000 Contingency Plan	High		

Hyperlinks were created to each document

In the master test plan hyperlinks were created to each supporting document, therefore creating one consolidated document for all project work products.

The testing strategy document included many items from the initial project scope through the above table that captured all testing and contingency information. The *Item #* column was used to uniquely identify the testing subject. The *Test Description* column identified what was tested, the *Responsibility* tracked who was responsible for

documenting and executing the tests. The *Status* column indicated whether the item passed, and listed each successful testing iteration, as well as the contingency plan. The *Priority* was listed to provide immediate testing focus on the area. High priority items were tested first, followed by the lower or lesser priority tests. This document became the all-inclusive report for the testing project.

## **Conclusion**

This testing strategy was created for a variety of reasons. One of them being that Liggett had gone through an extensive upgrade from 9.4 Character to 10.7SC and had little or no documentation to support the testing of the applications. Liggett wanted to develop a testing strategy that could be reused in the future to save them time and money. Well knowing that the application functionality would change the testing strategy was developed generically enough where it can be used for subsequent upgrades to Release 11 or 11i.

One of the most important aspects of the project was the well experienced project members, many of them had been using Oracle Applications since its inception back in 1993 and had an excellent working knowledge of the applications. Their experience with prior testing projects and the Oracle applications allowed them to create very effective testing documentation. Another key to the projects success was the assignment of a very experienced project manager who championed the testing project as well as provided functional expertise when needed. We also want to acknowledge the use of experienced consultants. Over the duration of the project we used three outstanding consultants who were able to successfully lead us through the testing project as well as resolve functional and technical issues as they occurred.

The development of a long-term testing strategy for your organization can be achieved through the use of tools like those found in this paper. Many, many times organizations find themselves designing and developing new testing documentation over and over again. Organizations tend to develop documentation on current projects that is very similar or the same as previous projects. The approach and development of the aforementioned documents are very typical in testing projects. There may be several ways to develop a testing strategy, but this is the approach that Liggett has taken and they plan to reuse it in the future. Since Liggett has been an application user for almost seven years and will continue to be an Oracle customer, they figured it was about time to start developing reusable work products, leveraging their investment.

Overall the creation and use of this testing strategy and approach was embraced by the organization and will be used for testing upgrades to future releases of Oracle Applications.