

LMS Application Project Summary

\$Revision: 1.3 \$

by M. Conneen

Table of contents

1 Business Objective.....	2
2 Project Overview.....	2
3 Solution.....	2
4 Solutions Architecture.....	2

Warning:

Our Confidentiality Agreement prohibits the sharing of detailed project information

1. Business Objective

Provide application interfaces and administration for the clients Learning Management System. This included such tasks as [developing the single sign on](#), developing the PeopleSoft interface, installing and configuring the LMS, developing and installing various third party interfaces, install, configure and tune IBM WebSphere.

2. Project Overview

Our client, a holding company of numerous Gas Local Distribution Companies with over \$20 billion in assets and revenues in excess of \$6.031 billion, had purchased a Learning Management System (LMS) to facilitate the management of such corporate training records as CDLs, OSHA Compliance Training, Corporate Directives, etc. Further, the client was in the process of migrating to a single PeopleSoft HR solution to house all employee records. Also, the client desired the ability to maintain all network access in their MicroSoft Active Directory deployment but did not want the LMS bound to any particular AD (or PeopleSoft) implementation.

Our project objective was to manage the installation and configuration of the IBM WebSphere application server as well as install and configure the application. Further, we developed various *Third Party* interfaces to integrate the LMS with various client business partners.

Finally, we interface with the clients ENOC to provide both IBM WebSphere and application administration. To assist in *first response* activities, we enhanced our [Simple Network Operation Center](#) objects to provide such services as ping alive, http alive, and other WebSphere monitoring type of functions.

3. Solution

To accomplish the objective we worked with the client's Business Analyst and selected LMS vendor to ensure the application was installed and configured properly. All *Third Party* interfaces were written in Java™ and deployed to Windows 2000 servers as Windows Scheduled tasks.

4. Solutions Architecture

Technologies used to complete project objectives were:

- JavaTM Version 1.4.2 Mac OSX 10.3
- JavaTM Version 1.4.1 Windows NT, Windows 2000
- JavaTM custom objects
- IBM WebSphere 4.x
- JUnit
- Groovy
- OOA/OOD
- UML
- Argo UML
- XML
- CVS on Linux
- Apache Ant, Maven
- Eclipse
- Oracle 9i