

Mark To Market Project Summary

by Mike Conneen
\$Revision: 1.3 \$

Table of contents

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

Warning:

Our Confidentiality Agreement prohibits the sharing of detailed project information

1. Business Objective

Develop a *Mark To Market* service for pricing various Gas Local Distribution Company (LDC) deals.

The project had the following stated goals:

- Create a central service to calculate Mark To Market per Deal.
- Create a client process to assist the client applications in enabling communication to the centralized server.
- Update a centralized operational data store.

2. Project Overview

Our client, a holding company of numerous Gas Local Distribution Companies (LDC) with over \$20 billion in assets and revenues in excess of \$6.031 billion, had initiated a *Corporate Credit Management* (CCM) project to assist in managing credit risk. Correctly pricing and rating current and future business partner gas deals was a project *Critical Success Factor!* (CSF).

As [Information Integrators, Inc.](#) had developed and deployed the client's enterprise pricing service, we were engaged to write a web service that would consistently and accurately price the client's complete portfolio of Gas term purchases (for ten LDCs) and feed that information to the enterprise Corporate Credit Management system.

3. Solution

Working with several Business Analyst across a geographically dispersed area, we first produced a *Could Be* model supported by various UML artifacts (Class Diagram, Component Diagram, and Use Cases).

Next, an XML-RPC web service was defined and created to interface with the enterprise pricing service. This web service provided the mechanism to consistently and accurately price each current and forward gas deal.

Finally, *stub client interface* components were created as samples for the various project to use as jump-starts for communicating with the XML-RPC based web service. While the examples were provided using the Java™ programming language. Some application

interfaces were developed using MicroSoft's dotNET technology.

4. Solutions Architecture

Technologies used to complete project objectives were:

- Eclipse
- JavaTM Version 1.3.1 Mac OSX 10.2
- JavaTM Version 1.4.1 Windows NT, Windows 2000 Server
- JavaTM custom objects
- OOA/OOD
- UML
- Argo UML
- XML
- XML-RPC
- Apache Ant, Apache Maven
- Tigris Scarab
- DB/2