

OPEN XML COURT INTERFACE

OXCI Development Iteration 3 – Test Report

June 29, 2004



MTG Management Consultants, L.L.C.
1111 Third Avenue, Suite 2700
Seattle, Washington 98101-3201
206.442.5010 206.442.5011 fax
www.mtgmc.com

I. INTRODUCTION

This document is intended to provide a report on the outcome of the test phase associated with the third development iteration of the Open XML Court Interface (OXCI) Electronic Filing Manager (EFM) project. It will describe what was tested and report the results of those tests. The tests will be described in detail. Differences between the tests associated with Iteration 2 and Iteration 3 will be addressed. A general overview of the test results will also be provided.

II. RELATED DOCUMENTS

References to the following documents may be found within this test report.

Title	Date	Source
OXCI EFM Design Document	Jan. 2004	counterclaim.com, inc., MTG Management Consultants, Inc.
OXCI EFM Test Plan	Mar. 2004	counterclaim, MTG
OXCI EFM junit HTML Iter. 2 Report	Apr. 2004	counterclaim, OXCI EFM
OXCI EFM Iter. 2 Test Report	Apr. 2004	counterclaim, MTG
OXCI EFM junit HTML Iter. 3 Report	June. 2004	counterclaim, OXCI EFM

III. DIFFERENCES BETWEEN ITERATION 2 AND ITERATION 3

During the third development iteration, the structure of the EFM and its packages had minor changes. Some of this was due to the requirements involved in ensuring that the EFM can execute in a Java 2 Enterprise Edition (J2EE) Web application environment. The majority of J2EE-related changes in this iteration had to do with Enterprise JavaBeans (EJBs) and using a relational database. Other changes were constituted in order to provide a more adequate layer of abstraction in the application. Specifically, some of the functionality implemented in the transceiver layer has moved into a new package titled "transport." Due to the changes of the structure in the application, the unit tests associated with the EFM also changed. The total number of unit tests associated with the code base at the end of the third development iteration was 189, 6 more than the tests associated with the code base at the end of development Iteration 2. The differences involved in testing the modified packages will be described in following sections.

IV. TESTS CONDUCTED AND THEIR OUTCOMES

Each of the packages described in the OXCI EFM Test Plan is listed below. Typically these packages represent modules defined in the OXCI EFM Design Document. The test cases associated with each package are described. The current status of the test case results is also discussed. Any changes in the testing structure between the second and third development iterations will also be outlined.

A. Admin Package

This package is essentially the same as during Iteration 2. No new test cases have been added. Some tests were modified to be more thorough. The total number of tests for this package is 102.

The admin package contains objects related directly to the user interface. These objects are divided into three categories: Actions, Forms, and Virtual objects. The objects generally conform to the Struts and Tiles-based Web application guidelines. A description of Struts and Tiles-based Web applications is in the OXCI EFM Design Document.

The Actions objects represent the application logic that is executed when some sort of action takes place in the user interface. An example of an action might be clicking on the submit button of a form in the application. The action associated with it will read the form data and make any core system component calls that it requires to decide what to do. It will then do what it feels is appropriate and set the scene for the next user interface that will be displayed. Currently, eight different Actions objects contain the application logic for the EFM. The Forms and Virtual objects are rather simple. They merely hold data to be used by the actions.

The Forms objects can also ensure that the data they contain meets specific conditions. Forms objects are created when an HTML form element is used by the user interface.

The Virtual objects are used by the actions to get data from previous actions and to store data for future actions to reference.

Since the package is rather complex, the test suite for it is also. The Forms and Virtual objects are rather simple to test. Mainly, the test cases associated with them simply check the data access functionality of the objects. The test cases for the Actions are much more complex. They must construct a similar Web-like environment for the action to execute in.

Currently, there are 102 test cases evaluating the performance of the admin package. Thirty-five tests are associated specifically with actions. The remaining 67 tests are run on the Forms and Virtual object classes. Currently, all tests execute successfully except one test associated with Actions, which facilitates document download events. This test fails due to incomplete data being made available to the action through the filing object. This data is available only in specific versions of the OXCI Schemas. The newer versions of Schemas have been used in development and have worked successfully with this test case. At the moment, the newer version of the Schemas has not been checked into the project's Concurrent Versions System (CVS) repository. This is causing the test to fail. The new versions of the Schemas will be checked in, and a new minor release will be issued. This failure will be addressed in the upcoming minor release.

In summary, 102 test cases function on this package. One failure is reported, and the reason for this failure has been identified. This failure will be addressed in a future code update.

B. Billing Package

Three tests are provided for the billing package. The first test verifies the integrity of the Billing Response object. The tests ensure that data can be safely stored and retrieved from the object. The other two test cases deal with the specifics of the VeriSign Billing module. The data storage methods are tested in one case, and the processing functionality is verified in a separate test case.

One test is currently failing in this package. The failing test checks that actual transaction processing occurs with the VeriSign PayFlow Pro servers. The failure is due to the fact that the latest OXCI payment Schemas have not been checked into the CVS repository. The tests have worked internally and should work after an upcoming code update.

C. Client Package

Three client objects are implemented in this package: an HTTP filing client and Simple Object Access Protocol (SOAP) and ebXML clients. The client objects are intended to be used by Electronic Filing Service Providers (EFSPs) that wish to easily file into the EFM. Five test cases ensure the proper function of these clients. The HTTP client test verifies that an HTTP submission can be successfully completed. The SOAP and ebXML objects are each more complicated than the HTTP filing client. The test cases associated with each of these clients ensure that data accessing

methods all function correctly and that a filing submission can be successfully completed.

One test case results in failure. The failure is due to the fact that the latest OXCI payment Schemas have not been checked into the CVS repository. The tests have worked internally and should work after an upcoming code update.

All other test execute successfully.

D. Config Package

The main objects of this package are the EFM Rule Set and the EFM Configuration object. Two tests ensure the proper function of these implementations. One test ensures that the rule set can be instantiated and passed on to the configuration object. The other test ensures that given the rule set, the configuration object executes correctly.

Both tests currently execute successfully. There has been no change in the testing of this package during the third development iteration.

E. Connector Package

The status of this package has changed during development Iteration 3. Prior to the third iteration two tests existed. With the new SOAP CmsConnectors, three new tests have been added.

The three new test cases deal with the functionality associated with initiating a SOAP Case Management System (CMS) connection. They also test the functionality of a servlet designed to be a call-back for the SOAP connection. Currently, two of the SOAP CmsConnector test cases fail. This is because the remote CMS side of the connection is not simulated. Therefore, the connection can not be established.

Two CmsConnector classes are provided which implement a Java Remote Method Invocation (RMI) connection. A test case is provided for each object. The remote object test case ensures that the data accessing methods behave properly. The local Java RMI test case attempts to make a connection to a remote object and submit a filing.

Currently, only the RMI test for the remove object and the SOAP call-back servlet execute successfully. The RMI local object test results in an error due to the fact that an RMI environment is not set up for it at test time. The proper environment could be

constructed; however, the environment would change for each machine (IP address) that the tests will be run on. So if it were to be modified to run successfully on counterclaim's test server, it may not run correctly on any other machine. The same is true for the SOAP test cases which fail to execute.

To summarize: two tests execute successfully, and three tests encounter an error condition.

F. Core Package

The core package provides implementations of objects required by the EFM for core functionality. These objects represent entities such as the server and the individual components of the system. The base class for a component has a single test case that ensures the data accessing methods function correctly. The EFM Controller object has three unit tests associated with it. One unit test was added during this development iteration. The Server object contains three tests. One test ensures the data accessors, another test verifies that the object can be instantiated correctly, and finally a test ensures the proper function of the server controls (such as starting and stopping).

The status of this package has not changed between development iterations. Currently all tests except one reach successful outcomes. The test that reaches an error is the test that processes filings in the controller. The current test sets up the SOAP CmsConnector. However, since the SOAP connector is not configured in the test environment, this test encounters an error.

G. Court Policy Package

This package provides implementations of the court policy object along with classes used to host and retrieve court policies. There are five test cases associated with this package. The simplest test case checks to ensure that the data stored in the configuration file regarding the court policy is successfully parsed and loaded in the court policy object. Two tests ensure that the court policy file can be hosted locally in the Web environment and can be read from a remote or local location. A fourth test ensures that the XPath queries executed on the court policy XML are correct. A fifth test case ensures that a filing validates against the policy rules.

All five of these tests currently execute correctly. The validate method was added to the court policy module during the second development iteration and is used by the controller class to ensure that incoming filings meet the requirements outlined in the policy.

H. Logger Package

The logger package provides a simple file system logger implementation. This module is used by the system when key events happen. A single test case verifies that the method in the logger object executes without error.

Currently, this test case executes successfully. This package has not changed during the prior development iteration.

I. Validator Package

The validator package handles the validation of XML instance documents. There are objects in the package that represent a validation problem and a handler for the problems. A single test case exists to ensure the data accessing methods of the validation problem. A single test case exists which verifies the proper execution of the methods contained within the problem handler object. Two test cases exist for both the Document Type Definition (DTD) validator and the XML Schema-based validator. One test ensures the data accessing methods of the validator classes, while the other test ensures that the DTDs or schemas can successfully validate.

Currently, six tests exist for this package, and one test case fails. The failure is associated with the DTD validation class. The failure results from the fact that we have configured the system to use the OXCI Filing Schemas, not the prior implementation's DTD files. This failure is expected and should not inhibit proper operation of the OXCI EFM.

J. Model Package

The model package is large. It contains objects that represent data used by the EFM. These objects consist of items like filings, documents, EFSPs, courts, etc. The package also contains managers for these objects that handle the creation, retrieval and persistence of the objects.

Currently, there are 35 separate test cases for this package. Many of the test cases ensure that the proper data accessing methods function correctly. A number of tests ensure that the manager objects successfully create and persist data objects. In general, application logic code is not stored in this package.

Thirty-four tests currently execute successfully. One test results in an error. The failing test is associated with a sample filing from a prior implementation of the EFM.

The prior filing is a Court Filing 1.1 document. During the second development iteration, proper execution of OXCI Schema-based filings was the focus, and a few test cases related solely to Court Filing 1.1 documents have resulted in error conditions. This should not affect the OXCI project.

K. Notification Package

The notification package provides a module capable of sending e-mail messages. One test case is provided with this package. The test case ensures that a valid e-mail message can be sent. Currently the e-mail message is sent to *openefm@counterclaim.com*, and the test executes correctly. This package has not changed during the last development iteration.

L. Security Package

The security package contains simple classes for using a login name and password validation for EFSPs. A security violation object is also provided. Two test cases are available for this package. One test case ensures the data access methods of the security violation object function correctly. The other test case ensures that the functionality of the simple security module works properly.

Currently both test cases result in successful outcomes. No changes have been made to this package during the third development iteration.

M. Transceiver Package

The transceiver package contains classes that accept a court filing and pass the filing on to the core EFM components. A new transceiver is implemented for each protocol used to pass a filing to the EFM. Currently three protocols are supported. These are HTTP, SOAP, and ebXML. Test cases exist for each implementation. The test cases ensure that any data accessing methods function correctly. The test cases also ensure that the transceiver objects receive filings and that these filings are properly passed on to the core EFM components.

There are five unit tests associated with this package. Currently, they all execute successfully.

N. Transport Package

The transport package was added to the EFM during the third development iteration. It provides a layer of abstraction around the messages that are being sent over various protocols. Some of the functionality found in the transport package was in the transceiver package during the last iteration.

Currently, there are four unit tests that ensure proper functionality of this package. They all execute correctly.

O. Util Package

The util package contains helper functions for the rest of the application to use. The methods are executed independently from any system-specific task. They are used by other code modules and are isolated from the modules that use them since many different modules may include references to these objects. Currently, they mainly provide Input and Output (IO) routines associated with file access.

Three test cases are provided for the util package. These task cases ensure the proper execution of the individual IO routines. Currently, these tests all execute correctly. No changes were made to this package during the third development iteration.

V. OVERALL RESULTS

One hundred and eighty-nine test cases are currently implemented in the OXCI EFM. Of these test cases, 180 execute successfully. Eight errors are reported, and one failure is reported. All of the errors or failures are expected. Most have to do with DTD- and Schema-related tests. Some of these failures will be corrected in the near future; others have no bearing on the OXCI project. The other failures and errors generally have to do with testing of RMI or SOAP CmsConnectors. These are also expected due to the fact that the receiving side of the connection is not configured for these tests.