
FSA Integration Partner
United States Department of Education
Federal Student Aid



We Help Put America Through School



eZ-Audit - PEPS Integration
School File

Task Order #116

March 20, 2003

Revision History

Date	Name (Lotus Notes ID)	Description of Revision
03/21/2003	Ann M. Keast	Initial Creation
04/07/2003	Jennifer C. Sri	Text Changes

Table of Contents

Introduction	1
Purpose	1
Scope	1
Intended audience	1
Background	1
Overview	1
Design Considerations	2
Assumptions and Dependencies	2
General Constraints	2
Goals and Guidelines	2
Development Methods	2
Detailed Design	2
System Architecture	4
Subsystem Architecture: SMTP Server	4
SubSystem Architecture: ITA Logging and Exception Handling Framework	5
Detailed System Design	6
SFASmtpClient JavaBean	6
Interaction Diagrams	7
References	8
Appendices	9
Jar Files	9

Introduction

Purpose

This document provides a high-level summary and detailed technical design for a component of eZ-Audit's technical architecture: School File. The school data provided by PEPS is critical to the accuracy of the data within the eZ-Audit system.

Scope

The School File is a part of the larger technical architecture framework that will be briefly described. However, the scope of this document will primarily focus on the School File.

Intended audience

This document is intended for eZ-Audit operations team application programmers who need to understand the School File in order to load school data into the eZ-Audit database.

Background

In the past, PEPS has been the sole owner and distributor of school and submission data. With the advent of eZ-Audit, some data is still owned by PEPS and consequently eZ-Audit requires setup and interval data from PEPS. PEPS will also continue to own school data and will distribute this school information to several applications on a daily basis. EZ-Audit must receive and import this school file data on a nightly basis.

Overview

EZ-Audit integrates with PEPS for school data. EZ-Audit achieves this integration through updates in the form of a school file feed, which is a flat text file received through the EAI Bus. EZ-Audit will be responsible for updating its database with the new or modified school data received from the PEPS school file.

On a daily basis, PEPS will provide a complete text file of all school data. This file will be available to the EAI bus, which is responsible for distributing this file as necessary to associated applications. EAI will put this file to a specified directory for eZ-Audit. Once in this directory, eZ-Audit will retrieve the modified rows of data and update its database tables accordingly.

Design Considerations

Assumptions and Dependencies

It is assumed that the EAI team will create a script that will be called once they have received the full school file from PEPS. The script that EAI will call an eZ-Audit-created script. This eZ-Audit script will be responsible for setting the classpath, making the java call (schoolFileLogging), and archiving the school file and error logs that are created. It is also assumed that eZ-Audit will create error logs and other info logs using ITA-created logging and will email these error logs using ITA's email module.

General Constraints

The Email Framework will require that a Simple Mail Transport Protocol Server (SMTP) is available so that emails generated by the client application may route mail to it. Currently ITA uses **sendmail** that is available on Sun Solaris.

Goals and Guidelines

The goal is to create scripts and a java program that will read in the PEPS full school file and update each record in the eZ-Audit database accordingly. The process should occur nightly and automatically, meaning it should be scheduled to run at a specific time each night the school file is received.

Development Methods

The java program and script were written independent of the eZ-Audit application and will be executed independent of the system. The java program was written and compiled using JDK 1.2.2, the JDK version available on the application server.

Detailed Design

The following is the detailed design for the program that will handle reading the file and updating the

eZ-Audit database tables accordingly:

PEPS School File Feed

Open PEPS school file

Create 01Array[x][y]: x is max number of records, y is number of fields updated by this type of record

Create 02Array[x][y]: x is max number of records, y is number of fields updated by this type of record

- Separate array for President and FAA contact types may be created

Create 04Array[x][y]: x is max number of records, y is number of fields updated by this type of record

Create 07Array[x][y]: x is max number of records, y is number of fields updated by this type of record

- Separate array for Primary and Secondary Accrediting Orgs may be created

Create count01, count02, count04, and count07 and set all to zero

Start reading records (one record at a time)

Read first record - skip 00 record

Start Reading 01 records

For each 01 record

store the OPE ID, SCHOOL NAME, LINE1 ADR, LINE2 ADR, CITY, STATE, PROVINCE, COUNTRY, ZIP, SCH TYPE, REGION CD, and CASE TEAM CD into the 01Array

increment count01

Start Reading 02 records

For each 02 record

If contact type is president

store the OPE ID, concatenation of President name, and email address into President's Array02

increment president count02

else if contact type is FAA

store the OPE ID, concatenation of FAA name, email address, phone number and fax number into FAA's Array02

Increment FAA count02

Skip 03 records

Start Reading 04 records

For each 04 record

store the OPE ID and DUNS number into the 04Array

increment count04

Skip 05 and 06 records

Start Reading 07 records

For each 07 record

If accrediting type is primary

store the OPE ID and primary accrediting org into primary 07Array

increment primary count07

else if accrediting type is secondary

store the OPE ID and secondary accrediting org into secondary 07Array

Increment secondary count07

Start reading any other records – skip and end reading in of records

Close file

PEPS School File Feed (continued)

Open connection to database

For (i=0; i < count01; i++)

update institution set column_name=01Array[i][2]... where ope_id = 01Array[i][0]

For(j=0; j < count02; j++) - do twice, president and FAA

update institution set column_name=02Array[j][2]... where ope_id = 02Array[j][0]

For (k=0; k < count04; k++)

update institution set column_name=04Array[k][2]... where ope_id = 04Array[k][0]

```
For(m=0; m< count07; m++) – do twice, primary and secondary  
    update institution set column_name=07Array[m][2]... where ope_id = 07Array[m][0]  
Close connection to database
```

PEPS will receive an error log each time the school file program is executed. These error logs will be emailed to a member of the PEPS team for review. A summary of the number of errors of each type will appear at the end the log file. The following is the list of the errors that may appear in the log file (associated OPE ID that caused the error will be identified if applicable).

- 01 - Detail Record (Title IV schools missing School Name)
- 01 - Detail Record (Title IV schools missing City)
- 01 - Detail Record (Domestic Title IV schools with an invalid State)
- 01 - Detail Record (Title IV schools missing Zip)
- 01 - Detail Record (Title IV schools missing Region)
- 01 - Detail Record (Title IV schools missing Institution Type)
- 01 - Detail Record (Title IV schools missing Case Team)
- 01 - Detail Record (Title IV schools missing Title IV Eligible Date)
- 01 - Detail Record (Title IV schools missing both Province and State)
- 02 - Contact Record (President - Type 34) Duplicates
- 02 - Contact Record (President - Type 02) Duplicates
- 02 - Contact Record (FAA) Duplicates
- 04 - Identifier Information Record (Title IV schools missing TIN)
- 07 - Accreditation Information Record (Primary - INS) Duplicates
- 07 - Accreditation Information Record (Primary - PRO) Duplicates
- 07 - Accreditation Information Record (Primary) Not Found
- Incorrect School File Version

System Architecture

The ITA Email Framework provides the ability to create, manipulate, and send Emails from any Java application running within a J2EE Application Server. Developers can use the simplified Email wrappers to read or send email or extend the JavaMail API's themselves for some very specific Email features.

Subsystem Architecture: SMTP Server

The ITA Email Framework requires that the framework have access to a Simple Mail Transport Protocol Server (SMTP). SMTP is the protocol that most Internet vendors implement to send emails across the Internet. The SMTP server is the actually workhorse that the Email Framework will connect to and forward emails that applications produce. Once the SMTP server has received the Email, it will connect to its forwarding partner (Another SMTP Server) and forward the Email out to the Internet. Currently the Email framework uses the SMTP Server that exists on the WebSphere machine that is provided via Solaris 2.6.

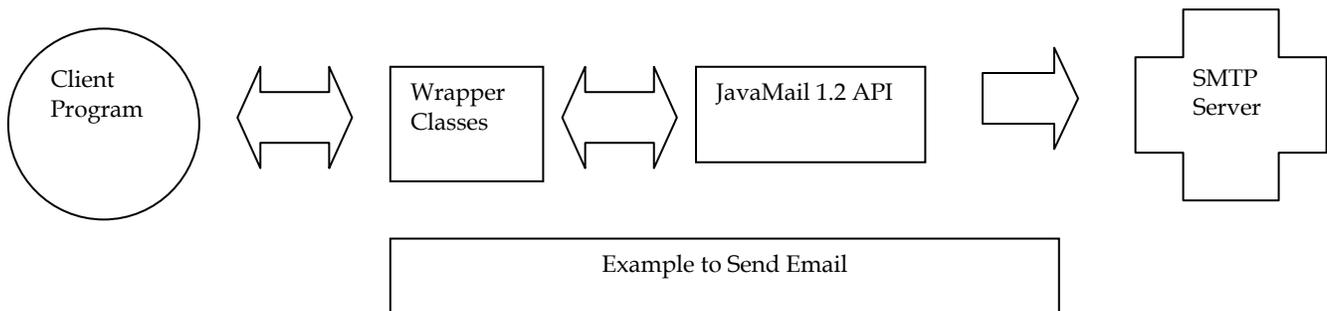
SubSystem Architecture: ITA Logging and Exception Handling Framework

The ITA Logging and Exception handling frameworks has been added to the Email framework to enhance debugging and tracing abilities. This should benefit operations ability to isolate problems that may occur within the framework. The Logging XML configuration document should be called rcs.xml and located in the home directory of the user making the java calls.

Detailed System Design

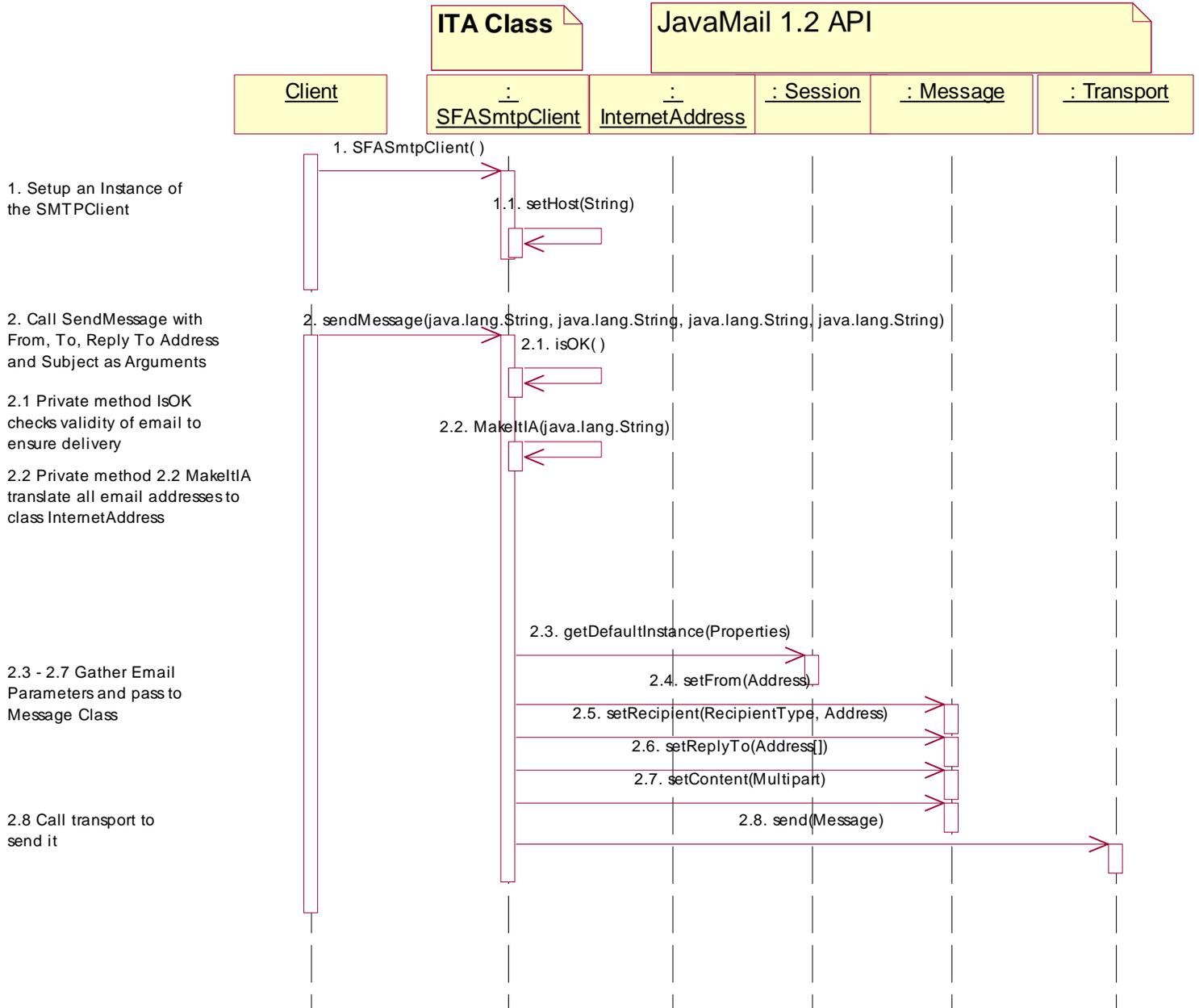
SFASmtpClient JavaBean

The ITA Email Framework includes a wrapper class that encapsulates the JavaMail API. This wrapper provide a simplified interface to the Sun JavaMail API and allow easy creation and manipulation of email traffic. The SFASmtpclient JavaBean is aimed at supporting Simple Mail Transport Protocol (SMTP), which is the predominate protocol for sending email. As specified in the Usage example a program needs to instantiate an instance of SFASmtpClient and then using the public methods build and manipulate and send the email.



Interaction Diagrams

This sequence illustrates the interaction between a client object and SFASmtpClient class to build and send a email to a SMTP Server



References

Java Activation Framework JavaDocs online

<http://java.sun.com/products/javabeans/glasgow/javadocs>

JavaMail 1.2 JavaDocs Online

<http://www.javasoft.com/products/javamail/1.2/docs/javadocs/index.html>

Jnet JavaBean Store

<http://www.java-shop.com/jnet.htm>

Appendices

Jar Files

The following Jars files are part of the ITA Email Framework and Logging Framework.

Mail Jar Files

Activation.jar
Mail.jar
MailApi.jar
Rcs_Email_01.jar
Pop3.jar
Smtplib.jar

Logging Jar Files

Jakarta-oro-2.0.1
Jdom-B6
Protomatter-1-1-5.jar
Utility.jar
Xerces.jar
Xml.jar

Logging XML Document

Rcs.xml