



Analytical Review Procedures using Web CAAT 2.97

Data analysis made easier

Web CAAT

Analytical review procedures are widely recommended as an efficient and effective audit technique. Many of these procedures can, and should be, automated to the extent possible. This is due not only to the often large transaction volumes present, but also to the ready availability of transaction and other data in electronic format.

Although the benefits of automating analytic review procedures may seem apparent, the rate of adoption of automated procedures has been estimated to be as low as 30%. In some audits, there is simply little, if any, data available in electronic format. In other instances, the amount of data to be reviewed is sufficiently small that the auditor may feel confident in “eye balling” the data. However, in this monograph, an attempt will be made to show why automated procedures may be more efficient and effective than manual procedures, even for relatively small transaction volumes.

The software used to illustrate the concepts is currently being used to [teach auditing concepts](#), statistical sampling and data mining. EZ-R Stats, LLC is registered with the North Carolina State Board of Certified Public Accountant Examiners as a provider of Continuing Professional Education.

This software is a “work in progress” and is updated based upon comments and suggestions received from practicing auditors as well as class participants. All [comments and suggestions](#) are welcome. A number of inquiries have been received regarding use of the software for the most common audit functions done by the “non power” users. In fact, this is the primary intention for the use of this software.

The name “Web CAAT” is derived from the two key aspects of the software – 1) it runs as a web service either on a stand alone machine or as part of a corporate intranet, and 2) it is a computer assisted audit tool (CAAT). The primary advantages of the software are 1) only a web browser is required to run it (nothing to install on the client machine) and 2) it leverages powerful open source software such as MySQL to provide extensive audit capabilities. Advantages also include the ability to easily share data between auditors, and also enhance audit productivity by enabling most auditors to run standard, menu driven procedures without the need for lengthy learning cycles. This is done by providing over 100 “ready to run” audit procedures, all using standard web based forms that most auditors are familiar with.

Document History

Revision History

Revision Number	Revision Date	Summary of Changes
1.0	2-25-2011	Initial version
2.0	3-18-2011	Updated to better describe data import procedures.
3.0	4-1-2011	Added database backup and restore procedures.
4.0	5-31-2011	Electronic Audit Programs
5.0	6-15-2011	Continuous auditing using “cron”
6.0	7-12-2011	Conversion to “Server2Go”
7.0	2011-09-03	Simplified Data Conversion Procedures

Table of Contents

- 1 SYSTEM OVERVIEW..... 1**
- 1.1 Who Should Use It..... 1
- 1.2 Purpose..... 1
- 1.3 Scope..... 1
- 1.4 Intended audience..... 2
- 1.5 Hardware requirements..... 2
- 1.6 Software requirements..... 2
- 2 GETTING STARTED..... 3**
- 2.1 Key application areas and audit program steps..... 3
- 2.2 Overview of audit programs..... 3
- 2.3 Audit Directory structure..... 4
- 2.4 Setting up (importing an audit program)..... 4
- 2.5 Processing audit program steps..... 8
- 2.6 Developing or updating program step code..... 12
- 2.7 Maintaining the menu system..... 14
- 2.8 Uploading documents..... 14
- 2.9 Auditing transactions 16
- 2.10 The “starter” kit 16
- 2.11 Continuous auditing..... 17
- 3 ANALYTIC REVIEW TECHNIQUES..... 18**
- 3.1 Overview and general procedures..... 18

3.2 Trend lines.....	18
3.3 Outliers.....	18
3.4 Cross tabulations.....	18
3.5 Data Classification.....	19
3.6 Data Summarization.....	19
3.7 Tests for "impossible" conditions	19
3.8 Potential duplicate payments.....	19
4 CONTINUOUS AUDIT / STORED QUERIES.....	22
4.1 Maintaining queries / procedures.....	22
4.2 Setting up the schedule.....	26
4.3 Running the queries.....	27
5 DATA IMPORT PROCEDURES.....	30
5.1 Overview.....	30
6 APPENDIX.....	36
6.1 Starting the system.....	36
6.2 Stopping the system.....	36
6.3 Database sign-on.....	36
6.4 Database backup.....	36
6.5 Database restore.....	37
6.6 WebCAAT installation.....	37
6.7 Security.....	37
6.8 Database administration.....	37
6.9 Menu customization.....	39

[6.10 Setting up and scheduling program steps to be run.....40](#)

[6.11 Installation on Unix internet servers.....40](#)

[6.12 Questions, comments and suggestions.....41](#)

1 System overview

This document is divided into the following sections:

- Section 1 – System overview
- Section 2 – Getting Started
- Section 3 – Analytic Review Techniques
- Section 4 – Continuous audit / stored queries
- Section 5 – Data Import Procedures
- Section 6 – Appendix

1.1 Who Should Use It

Auditors, researchers, business analysts and academics who use data analysis to perform their jobs.

- Auditors: can use the software to for a variety of common audit tasks. Altogether, over 40 useful analytical audit functions are included
- Researchers: use the software for:
 - Data analysis, trend investigation
 - Preparation of statistical reports and charts

1.2 Purpose

The purpose of this monograph is to provide a practical guide to analytic review procedures using Web CAAT. The auditor does not need special computer skills in order to be able to perform these tests because they are largely menu driven with “fill in the blanks”.

Development of the software began in August 2005 when the author searched fruitlessly for a relatively easy to use, economical software package for analyzing data on Excel work sheets (and other). During its development, suggestions and improvements were made by a variety of audit practitioners.

More information about the system is available from the [website](#). More information is also available about the [author](#).

1.3 Scope

This guide provides a general approach for performing analytical review procedures, as well as examples of use.

1.4 Intended audience

The software is intended for use by both internal and external auditors, researchers, program monitors, students learning data analysis, business analysts and anyone else interested in performing data analytics in a more efficient and effective manner.

1.5 Hardware requirements

At least 512 MB of memory (more if possible). Minimum free disk space is 500 MB.

1.6 Software requirements

Requires Windows XP, Vista or Windows 7, Linux, or Mac OS/X. A web browser is required to use the software.

2 Getting Started

Most auditors will start by using the audit program steps which are pre-defined and included with the system. This can be done by importing key data from application systems of business processes being audited into the Web CAAT database. Once the data has been loaded, then the pre-defined program steps can be tailored to specific audit needs by specifying the names of the data elements, providing starting and ending dates for the audit period, etc.

Often the most challenging part of the process will be obtaining and importing data into the system. Section 5 of this manual provides more specifics on this process.

2.1 Key application areas and audit program steps

The starter system includes a number of pre-defined audit program steps which may be of use in the following application areas or business processes:

- Accounts Payable
- Fixed Assets
- Expense Reports
- Inventory
- Procurement Cards

For each of these audit areas, suggested program steps are provided. These steps can be tailored to meet specific audit requirements. In many cases, audit program steps useful in one area are also useful in other audit areas as well. It is a relatively simple process to replicate or re-use audit program steps.

2.2 Overview of audit programs

Web CAAT provides the ability to establish and use electronic audit programs. Electronic audit programs are similar to traditional audit programs in that they document audit planning, background information, audit objectives and audit procedures. However, they differ in that there is no single “document” which contains all this information. Instead, the information is maintained in a database on the Web server, which can be accessed through a web browser using an intranet or other connection.

For each audit step or procedure, work papers are completed and uploaded to the server. The auditor then indicates which steps have been completed, and they are then available for review. The most significant difference, however, is that certain audit steps which can be performed directly using computer assisted audit tools, using the information contained in the audit program (and stored in the database).

Getting Started

2.3 Audit Directory structure

In Web CAAT, audits are organized through the use of two categories, unit and audit number. A unit is an organizational or other logical grouping for the overall structure of the audit information. Some organizations may wish to use only a single unit, others may wish to establish a number of units, perhaps one for each subsidiary, client, geographical location, etc. Within a unit, one or more numbered audits may then be established. All the audit information is stored within a database on the server.

An example may clarify:

2.4 Setting up (importing an audit program)

Because an electronic audit program does not exist as a single document, the information comprising the audit program must be imported into the database using a text file in a special format and a menu item. The input document used to prepare an audit program is a simple text file. Each line of the text file represent different information, each of which is identified by a symbol on the following line:

Description	Terminating symbol
• An audit step	#
• An audit program caption title	#T
• An audit objective	#O
• An item of information (e.g. planning, background, etc.)	#I

These items are contained on one or more lines followed by the terminating symbol which identifies the type of information. There are examples of audit program provided.

The typical procedure to convert a word document or PDF document to electronic format is as follows:

1. Create a text file with the information from the audit program. (Can use html coding to format tables or other information which needs to be displayed in a special format)
2. Divide each item of information using the symbols above.
3. For program steps than can be performed electronically, define/develop the script which performs the process.
4. Import the text file into the system using the menu system provided (“Audit programs | Import audit program”).

The program file import form requests the following information:

1. Name of the text file which contains the audit information (must first be uploaded)

Analytic review techniques

2. The unit to which the audit program step relates (can copy audit programs if there is multiple use)
3. The assigned audit number
4. The short hand notation for the primary area of concern

An example form appears as follows:

The menu item is “Audit Programs | Import Audit Programs”

Sign In		
Audit Programs	› List Audits	root
Audit Documents	› Import Audit Program	test
Audit Scripts	› Upload Single PHP Script	localhost
Table Selection	› Upload Batch of PHP Scripts	file
Commonly Used	› Run script for an audit step	AP-pgm.txt
Date Functions		01
Numeric Functions		1234
Other Functions		Audit Title
Other Processes		Title of audit here
Data Import		Area of Concern
Configurations		AP
Risks		
Schedule		Process
Sign-Out		
About		

Load audit program from text file

Form instructions

For some forms, the **mouse over** will provide a description or purpose of the form shown in the 'Help' box just above the Process button. In particular, the information above should be completed as follows:

- o Select the name of the text file containing the data to be loaded
- o Provide an audit number which will uniquely identify the audit (letters)

Clicking on the menu item brings up the following form:

Getting Started

Sign In	Load audit program from text file	
Audit Programs >	Logged in as	root
Audit Documents >	Database selected	test
Audit Scripts >	Host	localhost
Table Selection	Load audit program from text file	AP-pgm.txt <input type="button" value="v"/>
Commonly Used >	Unit	01
Date Functions >	Audit Number	1234
Numeric Functions >	Audit Title	Title of audit here
Other Functions >	Area of Concern	AP
Other Processes >		<input type="button" value="Process"/>
Data Import >	Form instructions	
Configurations >	For some forms, the mouse over will provide a description or purpose of the data shown in the 'Help' box just above the Process button. In particular, the information above should be completed as follows:	
Risks >	<ul style="list-style-type: none">o Select the name of the text file containing the data to be loaded.o Provide an audit number which will uniquely identify the audit within (letters)	
Schedule >		
Sign-Out		
About		

The five items of information needed are as follows:

1. The name of the text file containing the audit program being imported. (Note that the file must be uploaded to the server.)
2. The applicable unit (default is 01)
3. Assigned audit number (can include letters)
4. Audit title – a brief description of the audit
5. Area of concern – an abbreviation for the area of concern (e.g. AP – Accounts Payable)

This process is then repeated for all audit programs to be imported. This information can be accessed using the menu item "Audit Programs | List Audits".

Analytic review techniques

Sign In	Unit	Audit Number
Audit Programs	List Audits	
Audit Documents	Import Audit Program	
Table Selection	Upload Single PHP Script	
Commonly Used	Upload Batch of PHP Scripts	
Date Functions	Run script for an audit step	Auditor:
Numeric Functions		
Other Functions		
Other Processes		
Data Import		
Risks		
Schedule		
Sign-Out		
About		

Click the “List Audits” item brings up the following form:

Sign In	Unit	Audit Number	Select
Audit Programs	01	1234	Accounts Payable process
Audit Documents	01	1235	Expense reports 2010
Table Selection	01	1236	Property records 2009
Commonly Used	01	1237	Inventory cycle
Date Functions	01	1238	Procurement Cards 2010
Numeric Functions	Id: root Table: fa Auditor: Im N Auditor		
Other Functions			
Other Processes			
Data Import			
Risks			
Schedule			
Sign-Out			
About			

Clicking on the audit title, brings up the electronic audit program.

Getting Started

Run Program Step			
Audit	1234		
Unit	01		
SQL	select seq, objective, steptype, auditor, completiondate from auditpgms where unit = '01' and auditno = '1234' order by seq		
Task number	Description		
Background			
Accounts Payable is a function of the City of Tampa's Accounting area and is organizationally located in the Division of Revenue and Finance. It is responsible for the payment of the City of Tampa's lawful bills.			
Statistics			
In FY 2001, the Accounts Payable Section processed 41,966 checks for payment compared to 49,562 in FY 2000. The decrease is attributable to a change in the methodology for remitting child support payroll deductions to the State. In FY 2001, the City no longer issued an individual check for each employee that had a payroll deduction for child support. Instead, one check is issued to the State on behalf of all employees that have child support payroll deductions.			
Planning the Audit			
Task	Procedure	Auditor	Completion Date
105	1. Print Quality Control Package and process Auditor Assignment and Independence Statement page.		0000-00-00
106	2. Prepare and send an entrance letter to applicable personnel in the area under review.		0000-00-00
107	3. Prepare a planning program and include a time budget for the preliminary survey work.		0000-00-00
	4. Conduct an in- house review of the following:		
108	1. Financial Data 2. Internal Policy & Procedures Manuals		0000-00-00

2.5 Processing audit program steps

Processing audit steps depends upon the type of step which can be:

1. A manual program step (e.g. interview, review of documentation, planning, etc.)
2. An information item (e.g. background information, audit objective, etc.)
3. An electronic audit program step, i.e. a step that can be accomplished using a script or other computer procedure.

For audit steps type 1 and 2, the process is very similar to that which auditors are already used to. Some sort of document is obtained, memo prepared, etc. This document is then uploaded to the server. Note that each audit program step may include one or more documents and the documents can be of various types such as Excel work books, Word documents, documents prepared using Word Pad, documents in Adobe format – in short almost any type of electronic document. Currently there is a limit on file size of 4 MB per document. (This limit may be increased if there is sufficient interest).

Processing of electronic audit program steps requires that computer processing scripts have been developed and uploaded to the database. The installed system contains over 100 examples of such scripts and they are stored in the application folders /server2go/htdocs/ap, /server2go/htdocs/er, etc. etc. Note that these examples are stored for illustrative purposes. The actual script used has been uploaded to the database.

There are at least two ways to run an electronic audit program step. The first is to select an audit program and then notice the task number shown to the left of the electronic audit program step. This task number

Analytic review techniques

can then be selected at the bottom of the form. Clicking on the selection brings up the first of several processing forms.

A second (and more direct manner) is to select the menu option “Audit programs | run script for a step”. This brings up a form which asks for the unit, audit number and task number to be run. Shown below is an example form:

Run single audit step script	
Logged in as	root
Database selected	test
Host	localhost
Unit	<input type="text" value="01"/>
Audit number	<input type="text" value="1234"/>
Taskseq	<input type="text" value="150"/>
	<input type="button" value="Process"/>
Home	Run another script
Form instructions	
To run a script, enter the following information	
<ul style="list-style-type: none">• Specify the unit number which will use the audit program• Specify the audit program number which will use the audit program• Specify the audit task step which will use the PHP script	

On this form, there are three items of information which must be provided – unit number, audit number and task sequence. The task sequence is obtained by examining the sequence number shown on the audit program list as shown below:

Getting Started

146	21. Determine if an annual security access review was conducted on a departmental basis for ADPICS and FAMIS.
147	22. Discuss any audit findings with the Audit Supervisor, Deputy Director and Audit Director. After receiving their approval discuss any findings with Auditee management.
148	23. Summarize and conclude.
Objective 6: Accounts payable master file has data integrity.	
Task	Procedure
150	AP01 - quantify invoices over zero
151	AP02 - same vendor number, invoice number, diff amount
152	AP03 - Check for duplicate invoices
153	AP04 - Run Benfords Law on invoice amounts
154	AP05 - Prepare a schedule of overpaid invoices
155	AP06 - List the 5 largest invoice amounts for invoices after 6-30-20

Once the unit number, audit number and task number are entered on the form, the “Process” button is clicked which brings up the starting input form used to gather the processing parameters. An example is shown below:

Electronic audit program step detail	
Document root	C:/xampp/htdocs
Unit	01
Audit	1234
Step	150
SQL	select phpshow, phprun, name from auditpgms where unit='01' and seq = '150'
OK	Fetches the requested documents
Show PHP	C:/xampp/htdocs/work/Show2898.php
File size - showform.php	1638
post php	tmp
Run PHP name	C:/xampp/htdocs/work/runap01.php
File size	1063
Run	<input type="button" value="Run Step"/>
Id: root	Table: fa
Auditor: Im N Auditor	Folder:

Clicking “Run Step” opens the following form:

Analytic review techniques

Accounts Payable - Step 1	
Logged in as	root
Database	test
Host name	localhost
Table	invoices
Column	Invoice Amount <input type="button" value="v"/>
Name of table containing invoice information	<input type="text" value="invoices"/>
Location to be tested	<input type="text" value="'ABC'"/>
Cutoff date (enter as YYYY-MM-DD)	<input type="text" value="'2004-06-30'"/>
Name of cost column	<input type="text" value="invoice amount"/>
Name of date column	<input type="text" value="invoice date"/>
W/P Doc	<input type="text" value="01/1234/Step-01.htm"/>
All columns names (for reference)	Invoice # <input type="button" value="v"/>
Example criteria	Show examples by clicking the drop down button to the right
Where (criteria)	<input type="text"/>

Information on the form can be changed as required, and then the “Process” button at the bottom of the form is clicked. This results in the performance of the audit program.

Getting Started

Report run on Monday 30th of May 2011 07:50:54 AM	
Auditor: Batch testing	
Table analyzed - invoices	
Comments - schedule of acquisitions - invoices	
Criteria specified - 'invoice date' > '2004-06-30'	
SQL used - select sum('invoice amount') as Total, count('invoice amount') as ItemCount, min('invoice amount') as Minimum, max('invoice amount') as Maximum, avg('invoice amount') as Average, std('invoice amount') as StdDev from 'invoices' as t where 'invoice date' > '2004-06-30'	
user- root	
Database - test	
Host - localhost	
Column - invoice amount	
This report - C:/xampp/htdocs/01/1234/Step-01.htm	
Rows processed - 1	
Totals	27,922,822.93
Count	5,055
Minimum	502.41
Maximum	10,497.28
Average	5,523.80
Standard Deviation	2,863.60
Debit totals	0.00

Work paper document on server

This report - C:/xampp/htdocs/01/1234/Step-01.htm

Results of audit procedure (upload or import into Excel)

2.6 Developing or updating program step code

To make permanent changes to an electronic audit program step, the source code of the audit program step should be updated using a text editor such as Word Pad, Scite, etc. Scite is recommended for those who do not already have a test platform for running PHP programs. The program is open source and more information can be located at <http://www.scintilla.org/SciTE.html>. Key advantages of Scite are that program code can be readily tested and developed using the editor, which provides information as to location of program errors as well as enables running and testing PHP scripts directly from within the text editor.

New audit program steps can be developed, usually starting with existing audit program steps and making the changes needed to provide new or different audit program functionality. The audit program steps are routines written in the PHP language (or called from routines written in the PHP language).

The starter system contains over 100 audit program steps as examples, all of which are written in the PHP programming language.

Analytic review techniques

Each electronic audit program step can generally be accomplished using two processing forms. The first is referred to as a “show” form, because it “shows” the processing parameters required for the procedure. Once the parameters have been input, the program then “posts” or provides this information to a “run” form which takes the processing parameters provided and does the actual analysis.

There is no formal naming convention required although the examples provided as part of the “starter” system have a naming convention as follows. The “show” form will be named YYNN.php. YY is the name of the process or subsystem e.g. AP for accounts payable. ER for expense reports etc. NN is a sequence or step number. The corresponding “run” program will then be named “RunYYNN.php” where the YY and NN correspond exactly.

As an example, the first script for the accounts payable audit program has a “show: script name of AP01.php and a “run” script name of “RunAP01.php”.

Scripts may be assigned to audit program tasks either individually (some tedious) or loaded in a batch of steps all at once. This requires that a text file be developed and uploaded. Several examples are provided and they are named BatchLoad-AP.txt, BatchLoad-ER.txt, etc. These text files of commands can then be processed using the menu item “Audit Programs | Upload batch of PHP scripts”.

Clicking the “Process” button will upload all the scripts to the database and provide a report of the results. If the script could be uploaded, the results will be reported in green, otherwise the results will be shown highlighted in yellow (generally the script was not found or the task number has not been established).

Batch post from text file	
Process Text	01, 1234, 150, c:/xampp/htdocs/ap/ap01.php, c:/xampp/htdocs/ap/runap01.php
Success	file c:/xampp/htdocs/ap/ap01.php successfully uploaded
Success	file c:/xampp/htdocs/ap/runap01.php successfully uploaded
Process Text	01, 1234, 151, c:/xampp/htdocs/ap/ap02.php, c:/xampp/htdocs/ap/runap02.php
Success	file c:/xampp/htdocs/ap/ap02.php successfully uploaded
Success	file c:/xampp/htdocs/ap/runap02.php successfully uploaded
Process Text	01, 1234, 152, c:/xampp/htdocs/ap/ap03.php, c:/xampp/htdocs/ap/runap03.php
Success	file c:/xampp/htdocs/ap/ap03.php successfully uploaded
Success	file c:/xampp/htdocs/ap/runap03.php successfully uploaded
Process Text	01, 1234, 153, c:/xampp/htdocs/ap/ap04.php, c:/xampp/htdocs/ap/runap04.php
Success	file c:/xampp/htdocs/ap/ap04.php successfully uploaded
Success	file c:/xampp/htdocs/ap/runap04.php successfully uploaded
Process Text	01, 1234, 154, c:/xampp/htdocs/ap/ap05.php, c:/xampp/htdocs/ap/runap05.php
Success	file c:/xampp/htdocs/ap/ap05.php successfully uploaded
Success	file c:/xampp/htdocs/ap/runap05.php successfully uploaded
Process Text	01, 1234, 155, c:/xampp/htdocs/ap/ap06.php, c:/xampp/htdocs/ap/runap06.php
Success	file c:/xampp/htdocs/ap/ap06.php successfully uploaded
Success	file c:/xampp/htdocs/ap/runap06.php successfully uploaded
Process Text	01, 1234, 156, c:/xampp/htdocs/ap/ap07.php, c:/xampp/htdocs/ap/runap07.php
Success	file c:/xampp/htdocs/ap/ap07.php successfully uploaded
Success	file c:/xampp/htdocs/ap/runap07.php successfully uploaded

Getting Started

2.7 Maintaining the menu system

The menu system is contained entirely within the module “cHtml.php”. The structure of the menu is simply an unordered html list which contains the contents of the menu item to be displayed with a link to the program to be executed.

2.8 Uploading documents

Audit documents can be uploaded using the menu item “Audit documents | Upload audit documents”.



The form is shown below:

Analytic review techniques

Upload audit workpaper document	
Logged in as	root
Database selected	test
Host	localhost
Unit	<input type="text" value="01"/>
Audit number	<input type="text" value="1234"/>
Taskseq	<input type="text" value="150"/>
<input type="text"/>	<input type="button" value="Browse..."/>
	<input type="button" value="Upload"/>
Home	Home
Form instructions	
To upload an audit document, enter the following information	
<ul style="list-style-type: none"> • Specify the unit number which will use the audit program • Specify the audit program number which will use the audit program • Specify the audit task step which will use the PHP script • If the script is a Show script or Run script , check the appropriate box • Browse to the location of the file to be uploaded and then click the "Upload" button. 	

Uploading a document consists of filling in the form, browsing to the document on the local system and then clicking the "Upload" button. Note that currently there is a size limit of 4 MB per document. There is no limit to the number of documents or type of document that can be uploaded.

When the document is uploaded, a confirming form is displayed.

Upload audit workpaper document	
Logged in as	root
Database selected	test
Host	localhost
Unit	01
Audit Number	1234
Sequence	150
File Name	FE_Payroll.doc
Temp Name	C:\xampp\tmp\php1F3.tmp
File Size	26112
File Type	application/msword
Success	file FE_Payroll.doc successfully uploaded
Upload	Upload another workpaper document
List	List uploaded workpaper documents
Delete	Delete another document
Approve	Approve W/P documents
Retrieve	Retrieve D/L W/P documents

Getting Started

Five options are shown on the form (but can also be obtained using standard menu items). Review and approval of an audit program step can be done both at the step level and for each document.

Note that the system does not enforce or restrict any access, so the review and approval can be done by anyone with an id and password to the system

2.9 Auditing transactions

Analytical review procedures are based on transactions which may have been summarized or not. Often the auditor is faced with too much data and too little “information”. Web CAAT, like other analytical systems, first needs to have the key transaction data loaded into the system before analysis can begin. Section 5 addresses the process for getting data loaded into the system.

The primary advantages of Web CAAT include:

- Pre-built functions for the most common audit tasks
- Significantly reduced time required to perform more complex extracts and analyzes
- No need to “pre-sort” the data
- Logging facility – log work performed, can be shared or used as a basis for future analysis
- Very few inherent limitations in terms of theoretical analytical capabilities
- Can be customized or tailored for use

The primary disadvantages of Web CAAT include :

- Database knowledge is required in some situations
- Data import procedures are not simple

2.10 The “starter” kit

The system includes example data for about ten commonly encountered business processes. Each of these processes also includes example analytic review procedures which may be used for testing/learning or as a start in developing analytic review procedures for any specific business process.

The starter kit includes analytic review procedures for the following processes or systems:

- Accounts Payable
- Fixed Assets
- Expense report

Analytic review techniques

- Inventory
- Duplicate Invoice testing
- Procurement Cards

2.11 Continuous auditing

Once analytic review procedures have been designed and tested, they can then be converted for use in a continuous auditing system. By this is meant that the procedures can be run on an unattended basis for the purpose of testing for exceptions, identifying trends and obtaining baseline data for audit or sample planning.

3 Analytic review techniques

The type of analytic review technique best used for any particular situation depends upon the nature of the data being tested. This monograph will provide examples and descriptions of procedures that are most commonly used.

3.1 Overview and general procedures

All of the procedures described can be completed using menus, drop down lists and “fill in the blanks”. For each type of analytical review procedure described, a general overview of the procedure and its purpose will be provided.

3.2 Trend lines

Much of the underlying data making up account balances can be effectively reviewed using trend lines. The basic concept of trend lines is fundamental. However, more advanced concepts, using statistical measures of regression and deviation can be quite useful in narrowing the audit focus to areas or transactions that are more likely to be in error or highlight underlying issues of interest. Making the analysis more difficult is that summarized data is either unavailable or time consuming to compile. This in turn makes it more difficult to isolate transactions or groups of transactions which may be statistically significant as to their differences.

3.3 Outliers

Extreme values are often of audit interest as they may constitute a significant portion of the amount being audited and may represent some of the key amounts or transactions comprising a balance. It is sometimes also important to determine if the amounts are sufficiently extreme to constitute “unusual” amounts given the inherent variability of all of the transactions being examined.

3.4 Cross tabulations

One of the more frequently used analyzes is the “cross tabulation” which is used to “drill down” to specific transactions of interest based upon criteria provided. Amounts measured are most commonly totals, but may also include counts, maximum and minimum values, averages and variability (as measured using standard deviation).

Analytic review techniques

3.5 Data Classification

Perhaps one of the most basic techniques of analytical review is to first classify data elements. Classification may be based on numeric values (e.g. stratified data), date values (e.g. ageing) or text values (frequency distributions).

3.6 Data Summarization

Before any analysis is undertaken, one of the first steps is to test the validity of the data by obtaining control totals which are then tied back to general ledger accounts, feeder system totals, etc.

3.7 Tests for "impossible" conditions

Another analytical procedure is simply basic tests of reasonableness. Invariably the auditor is aware of what ranges transactions may have, as well as reasonable values for both account totals and transaction details. There are many tests which can be applied to determine if the data being examined is in fact, reasonable. Often such tests are labeled as tests for "impossible" conditions such as over depreciation of fixed assets, payroll calculations which do not extend or cross foot, negative inventory counts, etc.

Analytical review procedures can be tailored to handle such types of tests on an automatic basis.

3.8 Potential duplicate payments

Despite the best built-in system controls, duplicate invoice payments are possible due to the human element required to key in the invoice payment information. What often happens is that vendors perceive that they have not been timely paid and submit a duplicate invoice. If the duplicate invoice is processed, and an error is made in keying in the information, e.g. a transposed invoice number, the system will generally not detect this and a duplicate invoice payment will be made.

Perhaps one of the most common reasons for duplicate invoice payments being made is that duplicate vendors have been established. A starting point for a review of payment processing should consider looking at controls over the vendor master and also doing tests to ensure that duplicate vendors have not been established. Such tests would include two or more vendors having the same IRS taxpayer identification number (TIN), mailing address, contact telephone number or bank account number.

Once tests over the vendor master have been completed, then tests for duplicate invoice payments should be performed. Tests for duplicate invoice payments can be performed by identifying two payments which are similar. Note that unless system controls are not working, it is unusual to encounter an exact duplicate on all key fields – vendor number, invoice number, invoice date and invoice amount. However, it is possible

Continuous auditing

to identify potential duplicate payments based upon matches of invoice payments using one or more columns of information. Matches for potential duplicate invoice payments can be classified as two major types:

- “Exact” match
- “Fuzzy” match

Exact match

An exact match is where the contents of one or more columns are identical for two payments. An example is two invoice payments for the same invoice number, same invoice date and same invoice amount (but possibly different vendor numbers). This example can arise when duplicate vendor numbers have been established.

It is also possible to have a match when certain information, e.g. invoice number, has been transformed. A common example is when embedded spaces are considered. Take the example of two invoices in the same amount having the same invoice date and from the same vendor. One invoice number “12345A” and the second invoice number is “12345 A”. Many invoice processing systems will consider these to be distinct invoice numbers. A similar situation can arise with case, e.g. invoice “12345a” and “12345A”. All of these invoice numbers can be first transformed before they are compared. Examples of types of transformations include:

- Removing all characters except letters and digits
- Converting all letters to upper case
- Examining only the left 4 digits
- Rounding the dollar amount to the nearest \$10.

Fuzzy match

Other instances of similarity, but not exact match, include **transpositions**. Few, if any, automated processing systems will check for transpositions, whether they occur in the invoice number, vendor number, invoice amount or invoice date. Transpositions are a special case of a more general test called “**Levenshtein distance**”. Levenshtein distance is a mathematical algorithm for measuring the extent of similarity and will can detect not only transpositions, but suffixes, prefixes, character insertions, deletions, etc. The primary disadvantage of this technique are that it can require substantial CPU processing. However, this disadvantage can be offset by using a continuous auditing technique, and doing the analysis as an unattended process, likely during “off hours”.

Analytic review techniques

Web CAAT has a number of examples of processes used for duplicate invoice testing. The examples are named DI01.php – DI06.php (DI – “duplicate invoice”).

There is no “canned” approach that will work in every instance. Instead, the auditor should try various techniques to see which is most effective in identifying potential duplicate invoices in a particular situation. Often these techniques can then be converted into a continuous auditing process which can be very effective.

Detail procedures for the detection of duplicate payments are located in a separate document.

http://ezrstats.com/Xampp/Duplicate_Payments_UserGuide.pdf

4 Continuous audit / stored queries

Once audit queries have been developed and tested and found to meet audit objectives, they can then be saved and run on a repetitive basis, possibly “off hours”. It then becomes feasible to have many procedures performed. Even though it may require significant processing, there is no direct auditor involvement, making the process efficient and cost effective.

As an example, the installation includes scripts to perform checking for duplicate invoice payments. These scripts can be run on an on-going basis and will detect potential duplicate payments whenever the payment file is loaded into the system.

In addition, it is also very feasible to continuously test for those “impossible conditions” that should never happen. Since the scripts can be run unattended and automatically, there is little or no cost in processing them during “off hours”. If they identify an exception, this information will be made available immediately.

4.1 Maintaining queries / procedures

Development of queries is most often done by starting with ad-hoc queries in order to obtain the results desired. Ad hoc queries can be performed using a two step process:

1. Select a “table” for processing, e.g. payroll
2. Select an analytical procedure, e.g. cross tabulations, Benford’s Law, etc.

All of this can be done using the menu system – “Table Selection” for the first step, and then selecting the particular analytical procedure for the second step.

Continuous audit / stored queries

Table selection

Sign In	Sign	
Audit Programs >	User id	root
Audit Documents >	Password	••••
Table Selection	Database	test
Commonly Used >	Host	localhost
Date Functions >	Auditor	Im N Auditor
Numeric Functions >	Audit Folder	Work
Other Functions >	Server path	C:/server2go/htdocs
Other Processes >	Sign-in	
Data Import >		
Risks >		
Sign-Out		
WebCAAT >	For some forms, the mouse over will provide help. The help shown in the 'Help' box just above the Process button above should be completed as follows:	

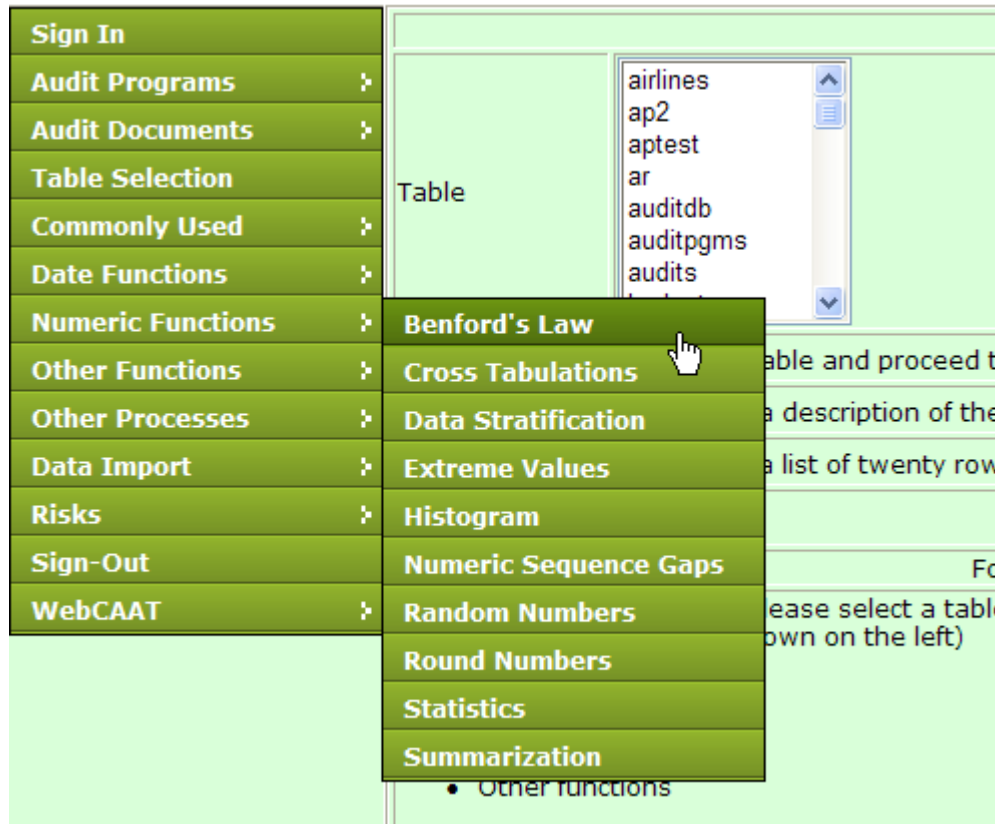
Select table

Sign In	Select table	
Audit Programs >	Table	<div style="border: 1px solid gray; padding: 2px;"> airlines ap2 aptest ar auditdb auditpgms audits budget </div>
Audit Documents >		
Table Selection	Main form	<input checked="" type="radio"/> Select table and proceed to main form
Commonly Used >	Describe	<input type="radio"/> Obtain a description of the table columns
Date Functions >	ShortList	<input type="radio"/> Obtain a list of twenty rows of the table for viewing the contents
Numeric Functions >	Process	
Other Functions >	Form instructions	
Other Processes >	To process ad-hoc queries, please select a table name from the drop down list. This is followed by the following menu items (shown on the left)	
Data Import >		
Risks >		
Sign-Out		
WebCAAT >		

Continuous audit / stored queries

Analytical procedure

Once the table has been selected, any of the 20 or so analytical procedures may be run using the menu system. For example to obtain basic statistics, select the menu item “Numeric Functions | Statistics”.



Once this is selected, a form is shown where the numeric column to be tested can be selected from a drop-down list. Once the column is selected, the “Process” button is clicked and the results are shown on the form as well as being written out to a work-paper file.

Select column

Continuous audit / stored queries

Sign In		
Audit Programs >	Table	payroll
Audit Documents >	Column	c
Table Selection	W/P Doc	c
Commonly Used >	All columns names (for reference)	CheckNumber
Date Functions >	Example criteria	GrossPay
Numeric Functions >	Where (criteria)	HolidayHours
Other Functions >	Purpose/Comments	Hourlyrate
Other Processes >		HoursWorked
Data Import >		PaidAmount
Risks >		SickHours
Sign-Out		VacationHours
WebCAAT >		System date population meas
	Help	
	Process	
		Form ir

For some forms, the **mouse over** will provide a desc shown in the 'Help' box just above the Process button above should be completed as follows:

View report

Continuous audit / stored queries

Population report	
Report run on Monday 7th of February 2011 09:07:04 AM	
Auditor: Im N Auditor	
Table analyzed - payroll	
Comments - Obtain basic measures of the population	
Criteria specified -	
Numeric column summarized - GrossPay	
Totals	2,409,106.55
Count	5,617
Minimum	-44.00
Maximum	3,032.47
Average	428.90
Standard Deviation	226.94
Debit totals	2,410,583.52
Debit Count	5,438
Credit totals	-1,476.97
Credit Count	143
Count of Zeros	36
Net totals	2,409,106.55
Counts	5,617
Powered by Web CAAT	

4.2 Setting up the schedule

The generally recommended approach to continuous auditing is to start small, get a few “wins” and then expand the scope of the effort. Often there are at least three hurdles to face:

1. Getting “buy in” from audit management as to the benefits from such a system
2. Getting resources to develop the scripts to be used in the process
3. Setting up a procedure to run these scripts in an automated fashion, usually during “off hours” when system resources are readily available.

Each of these hurdles may be overcome using the techniques summarized below:

1. Audit management “buy in” can often be achieved with a few “small wins” which can then in turn be expanded upon. Often auditors will have a pretty good idea as to good candidates for testing procedures which can produce “small wins” based upon recently completed audits and areas with known weaknesses.

Continuous audit / stored queries

2. Scripts can be developed with minimal technical knowledge by first running some queries in Web CAAT and noting the results. Any query in Web CAAT can be converted to a script just by replacing values in the examples provided. As a result, it is possible to set up some very useful test scripts without investing a lot of time and effort in training or computer program development.
3. Setting up a procedure to run these scripts “off hours” is also a relatively straight forward process. How this is done depends upon the environment in which Web CAAT is operating. If the environment is Windows, then there are two primary options – “Windows scheduler” and “PyCron”. The windows scheduler is provided by Microsoft and is included as part of Windows. Although it is effective, it is also fairly cumbersome and inconvenient to use. In the opinion of certain web hosting services and others (myself included) there is a much more effective option – “[PyCron](#)”. PyCron is also open source and is designed to work much like the traditional “Cron” process found in Unix. All of the syntax and processing is the same between PyCron and cron. There are also very good write-ups of PyCron and explanations of its use. Web CAAT also includes numerous examples in the folder named “/server2go/htdocs/cron”.

An article which provides an overview of the system is available at <http://www.bigbluehost.com/article4.html>.

The system can be freely downloaded at Source Forge <http://sourceforge.net/projects/pycron/>. License is GNU General Public License (GPL).

Installation procedures on Windows are relatively straightforward and the system can typically be set up in 15 minutes or less. The large majority of the time will be devoted to setting up all the continuous audit scripts to be run using the scheduling file named “crontab.txt”. Generally, a good starting point will be to take the schedule provided with Web CAAT and then tailor it to meet specific needs.

4.3 Running the queries

The queries are run using the Unix utility program “cron” or the Windows look-alike “PyCron”. The procedure consists of xx steps:

1. Install “PyCron” (download page is at <http://www.kalab.com/freeware/pycron/pycron.htm>)
2. Configure “PyCron” by editing the file in the installation folder, typically c:/program files/pycron.cfg, using a text editor such as WordPad to specify the location of the PyCron log and crontab.txt file. If Web CAAT is installed on the “E” drive, then the config file contents will appear as follows:

```
[pycron]
old_dow_compatibility = 0
crontab_filename =e:\xampp\htdocs\cron\crontab.txt
log_filename = e:\xampp\htdocs\cron\pycron.log
```

Continuous audit / stored queries

- Review the contents of the file “/server2go/htdocs/cron/cron.ini” which contains numerous examples of queries suitable for continuous auditing (these queries can be edited using a text editor such as Word Pad and changed to suit other needs). An example of a portion of this file which checks for over-depreciated fixed assets is as follows:

```
[task1]
;
; prepare a schedule of over depreciated assets
;
table=fa
auditor = Mike Blakley
column = cost
where = (cost - ad) < salvage
comments = prepare a schedule of over depreciated assets
title = Over depreciated assets
pgm=extract
outfile = /server2go/htdocs/cron/Reports/t-task1.htm
```

- Set up the entries in the “crontab.txt” file which was defined in step 2 above. An example to run the query above is shown below.

```
#
# task 1
# Obtain data extraction - over depreciated assets
12 1 * * * "c:\xampp\php\php.exe" c:\xampp\htdocs\cron\Extract.php [task1]
```

Any lines beginning with an # are comments and are ignored. The single text line that processes the script is more fully explained in the crontab.txt itself, however it is in industry standard “cron” format. The command can be interpreted as follows – at 1:12 a.m. every day run the program, “c:\xampp\php\php.exe” (which is standard PHP). The program to be run is the PHP program named c:\xampp\htdocs\cron\Extract.php . This program has a single argument, namely “[task1]”. This single argument is the name of the section in the cron.ini file that is included with the system.

- Review the report results. The existing script schedule includes a step which prepares a report named “Exception-Reports.htm” which lists all of the reports produced, provides the comment that was used when setting up the routine as well as a link to click in order to view the report. The report provides the date and time the report was produced, including file size.

Continuous audit / stored queries

Add/modify the scripts as needed. Note that some (but not all) of the ad-hoc queries automatically update the file /server2go/htdocs/logfile.ini which includes information on the commands run. This information can then be copied over to "cron.ini" and set up in the schedule.

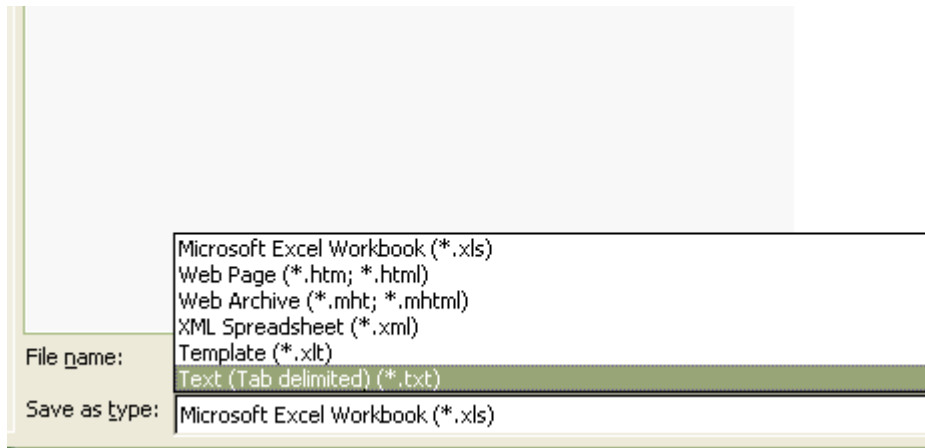
5 Data Import Procedures

5.1 Overview

There is a complete tutorial/overview of the data import process available at <http://webcaat.org/moodle/> which provides detail step by step procedures, video tutorials, exercises and quizzes.

Below is a recap of some of the information presented there.

Before data can be analyzed using the system, it must first be imported into the system. Imported data must either be in delimited text file format or converted into that format. The most common examples of “delimited text file” format are “comma separated values (csv)” or “tab separated values”. Almost all software programs, including Microsoft Excel and Access, provide for “exporting” data in formats such as tab separated values. For example, in Excel, a sheet can be saved in tab separated value format by selecting the output file format from the drop down list during the save operation (shown below).



There are two primary means to import data into the system 1) Basic (simple) import or 2) use standard database techniques. Each has their advantages and disadvantages:

	Basic (simple) import	Database standard
Advantages		
	Easy to use	Supports more sophisticated database structures
	Faster	Better control over the process
	Less knowledge required	
Disadvantages		
	Not able to handle all data	Learning curve
	Less control over conversion process	Slower process

Appendix

Basic (simple) import

The process for doing a basic import requires three steps:

1. Export data from Excel or other source in text delimited format
2. Import the data into the system using menus
3. If necessary, convert any date/time columns to the proper formats using menus

Database standard import

The process for importing data using standard database procedures is as follows:

1. Export data from Excel or other source in text delimited format
2. Define table structures to contain the data
3. Create the table(s)
4. Import the data using a menu system and database import procedures using “load data” command

Menus for basic (simple) import

To illustrate the process, a simple example will be shown. Data is contained on an Excel worksheet named “Inventory Data” and is to be loaded into the database with the same name (Inventory Data).

Data on the worksheet appears as follows:

	A	B	C	D	E	F	G	H
1	Inventory Date	Warehouse	Stock Numbr	Units	Unit Cost	Extended Co	Employee	Inventory Tag
2	6/30/2007	AB	5263	7	17.06	119.42	CE	1220
3	6/30/2007	C2	98795	504	4.25	2142	CE	1221
4	6/30/2007	C1	53984	616	5.86	3609.76	JF	1222
5	6/30/2007	FF	21312	117	3.02	353.34	CE	1223
6	6/30/2007	YY	12940	337	11.1	3740.7	JF	1224
7	6/30/2007	AB	58610	61	16.14	984.54	CE	1225
8	6/30/2007	C2	8823	54	3.01	162.54	JF	1226
9	6/30/2007	C1	63404	720	8.21	5911.2	JF	1227
10	6/30/2007	FF	28578	153	15.67	2397.51	CE	1228
11	6/30/2007	YY	30695	736	18.48	13601.28	JF	1229
12	6/30/2007	AB	39714	42	14.83	622.86	CE	1230
13	6/30/2007	C2	80582	413	11.15	4604.95	JF	1231
14	6/30/2007	C1	35623	414	10.71	4433.94	JF	1232
15	6/30/2007	FF	66074	340	10.66	3624.4	JF	1233
16	6/30/2007	YY	5885	178	15.8	2812.4	JF	1234
17	6/30/2007	AB	5904	8	2.03	16.24	JF	1235
18	6/30/2007	C2	31586	168	7.71	1295.28	CE	1236
19	6/30/2007	C1	42337	488	16.13	7871.44	CE	1237
20	6/30/2007	FF	5627	38	16.95	644.1	CE	1238

Appendix

Note that the work sheet contains labels on row one and does not contain any blank rows or blank columns. This is the format needed for a successful data import. The sheet is then saved as a delimited text file using the "Save As" function in Excel (the same principle applies for data exported from Access databases, Oracle databases, etc.)

File name:	Inventory Data.txt
Save as type:	Text (Tab delimited) (*.txt)

The process of loading data into the database consists of the following steps:

1. Convert the data to be loaded into a text file in delimited format. Most computer applications that data have a built-in facility for exporting the data from that application to a text delimited file. A text delimited file is a text file (for example a file that can be edited in Notepad, WordPad, etc.) where each of the column values are separated by a delimiter. Common types of delimiters include the following:

a. Comma separated values (CSV)

b. Tab separated values (TSV)

c. Other delimiters such as colon, pipe, etc.

2. Once the data has been converted to a text delimited format, update the text file to the server.

3. Determine where the data is to reside, i.e. which database table.

4. If the table does not currently exist, then use the form provided to change or specify the column names and the column types. Once all this information has been entered, click the "Process" button.

5. If the table already exists, i.e. additional information is to be added to the table or replace existing data, then the process is similar, with two exceptions:

1. The column types need to be verified for the columns of date or time format only

2. Whether existing data needs to be replaced, must be answered by selecting "Yes" or "No". Once this information has been completed, click the "Process" button.

When the data has been loaded into the table, twenty rows of data will be displayed in order to see how the data looks. Note that although the table may contain a considerable number of rows, only twenty rows of data are shown.

Menu item – “Data Import | Data Import”

Appendix

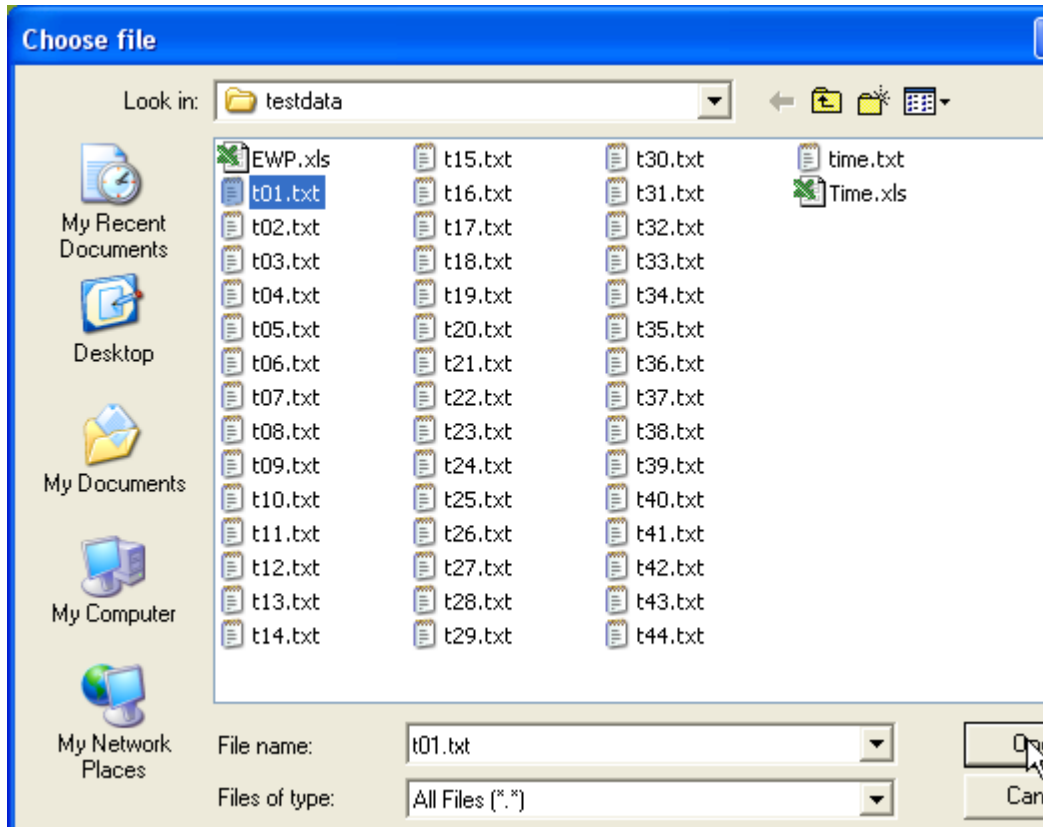
Sign In		
Audit Programs	Table	payroll
Audit Documents	Column	HoursWorked
Table Selection	W/P Doc	Work/Statistics.htm
Commonly Used	All columns names (for reference)	BirthDate
Date Functions	Example criteria	Show examples by clicking t
Numeric Functions	Where (criteria)	
Other Functions	Purpose/Comments	Obtain basic population meas
Other Processes		
Data Import	Data Import	
Risks	Create table using existing SQL statement	
Sign-Out	Drop table	
WebCAAT	Load data from text file	

Browse for file to upload

Upload a file			
Upload file	<input type="text"/>	<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>
Rows to scan to determine column types	<input type="text" value="100"/>		
Form instructions			
Importing data into the system is a multi-step process which begins with a delimited text file. The following means could have been created by any of the following means:			
<ul style="list-style-type: none">• Save an Excel sheet in tab separated value format• Export a table from an Access database• Export table data from an ERP or other database system• Convert a text report file into delimited format			
The first step in the process is to locate a text file in delimited format on the local drive and Web CAAT server (can be the same machine). To upload a file, click the "Browse" button and the location of the file to be uploaded, select it by double clicking, and then click the "Upload" button.			
Id: root	Table: payroll	Auditor: Im N Auditor	Folder:

Appendix

Select the file



Upload to server

Upload a file			
Upload file	C:\server2go\htdocs\testd; <input type="button" value="Browse..."/> <input type="button" value="Upload"/>		
Rows to scan to determine column types	<input type="text" value="100"/>		
Form instructions			
Importing data into the system is a multi-step process which begins with a delimited text file which could have been created by any of the following means:			
<ul style="list-style-type: none">• Save an Excel sheet in tab separated value format• Export a table from an Access database• Export table data from an ERP or other database system• Convert a text report file into delimited format			
The first step in the process is to locate a text file in delimited format on the local drive and Web CAAT server (can be the same machine). To upload a file, click the "Browse" button and the location of the file to be uploaded, select it by double clicking, and then click the "Upload" button.			
Id: root	Table: payroll	Auditor: Im N Auditor	Folder:

Appendix

Once the data has been uploaded to the server, there are two options:

1. Load the data into an existing table (if one exists), or
2. Create a new table.

The system will guess at the data types for each column of information in the uploaded file. However, this guess must be closely examined and changed, if necessary, by selecting the actual type of data from the drop down list of values. Once all the values have been selected, the “Process” button is clicked.

For the best explanation of the process, view a short video tutorial in the course “Data Conversion – Getting the data ready for analysis” available free at <http://webcaat.org/moodle> (Select “Login as a guest”). If you wish to take the quizzes or post in the forums, it is necessary to register (send name and contact info to Mike.Blakley@ezrstats.com – an id and password will be e-mailed within 24 hours).

6 Appendix

6.1 Starting the system

In order to start the system, the web services and database must be started. This is generally done by double clicking a shortcut to the program “Server2go.exe” which is found in the root directory of the installed system. Once the system is started, it is then available using the web server, i.e. anyone on the intranet, or on the local machine through the use of “localhost”.

Access using a LAN will generally be accomplished using a browser pointing to the IP address of the server, e.g. <http://10.271.3.35> (or using a host name if one has been established in the “hosts” file). For use on a local system, the command will be <http://localhost/> (without the quotes).

Starting up the system is generally not a lengthy procedure, lasting from 15- 30 seconds, depending upon the size of the server machine, database and number of files contained on the server.

6.2 Stopping the system

To safely shut the system down, click the “X” box on the upper right corner of the form. If the system is not properly shut down there is a risk of corruption to the database or loss of information entered into the system. Generally the shutdown process will not be a lengthy procedure.

6.3 Database sign-on

Almost all the applications on the server require that the user first be signed on to the MySQL database. If the user is not signed on, then a message “Please sign on to the database” will be displayed in response to every processing request.

Database sign-on consists of providing a valid user id and password. The default values provided with the system are “root” for a user name and “test” as a password. Almost always these values are changed (or should be changed) at the time of installation. Check with your system administrator for more information.

6.4 Database backup

Data can be backed up at any time by selecting the menu item “Other Processes|Database backup”. A form is presented which shows the folder where the data will be backed, as well as the location of the MySQL routine “mysqldump” which performs the backup. Generally, neither of these values will need to be

Appendix

changed. Results of the backup will be stored as a file with a time stamp in the file name to make it easier to determine when the backup was taken. The system will store the backup information both as a text file and also as a zipped file. Example name for the backup would be “bkup-2011-3-28-8-31-03.sql” for a backup taken on 3/28/2011 at 8:31:03 a.m.

6.5 Database restore

Once a database has been backed up, that information can also be restored using the “Other Procedures| Database restore” menu item. Information needed for the form is the name of the backup file. Note that generally backups will be stored in the folder “backup” unless an alternative location has been specified.

6.6 WebCAAT installation

The installation process consists primarily of downloading and unzipping the software. The installation does not require any system changes such as registry entries, processes, etc. It is entirely feasible to run the system from a removable device such as a portable (external) hard drive, flash drive, etc. Details for the procedure are described at <http://ezrstats.com/Xampp/home.html>.

More information can be obtained by taking the free online course described at <http://ezrstats.com/Xampp/Register.html>.

6.7 Security

Like any web server, there are potential security concerns, especially if the system is installed facing the Internet. However, in most environments, the system will be installed within a corporate LAN, or else on a stand alone machine that is not connected directly to the Internet.

The standard facilities of the Apache server may be used to password protect web pages on the server. Often the biggest security risk is the data contained within the database. This data can be protected through the use of MySQL user ids and passwords, along with standard database security design.

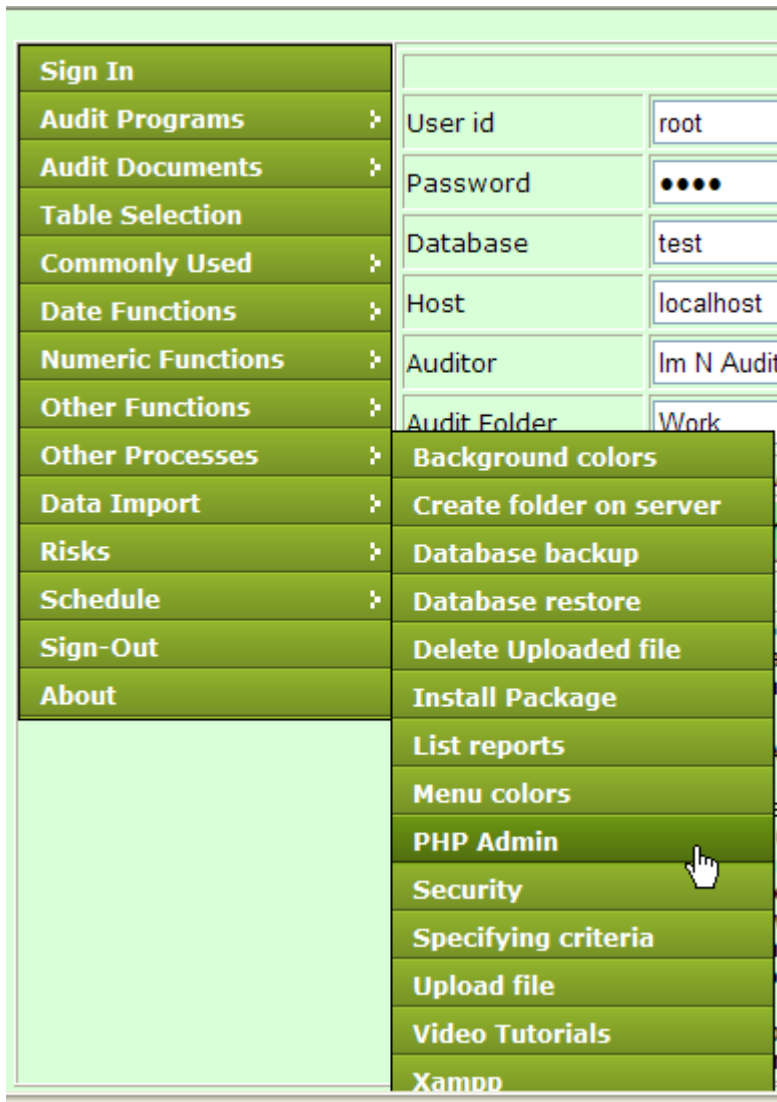
6.8 Database administration

If there is a need for database administration, such as establishing a new user, changing or resetting passwords, etc. this can be done using the built-in utility program “PHPAdmin”. This is an open source program for administration of MySQL databases using a PHP platform. The system contains user

Appendix

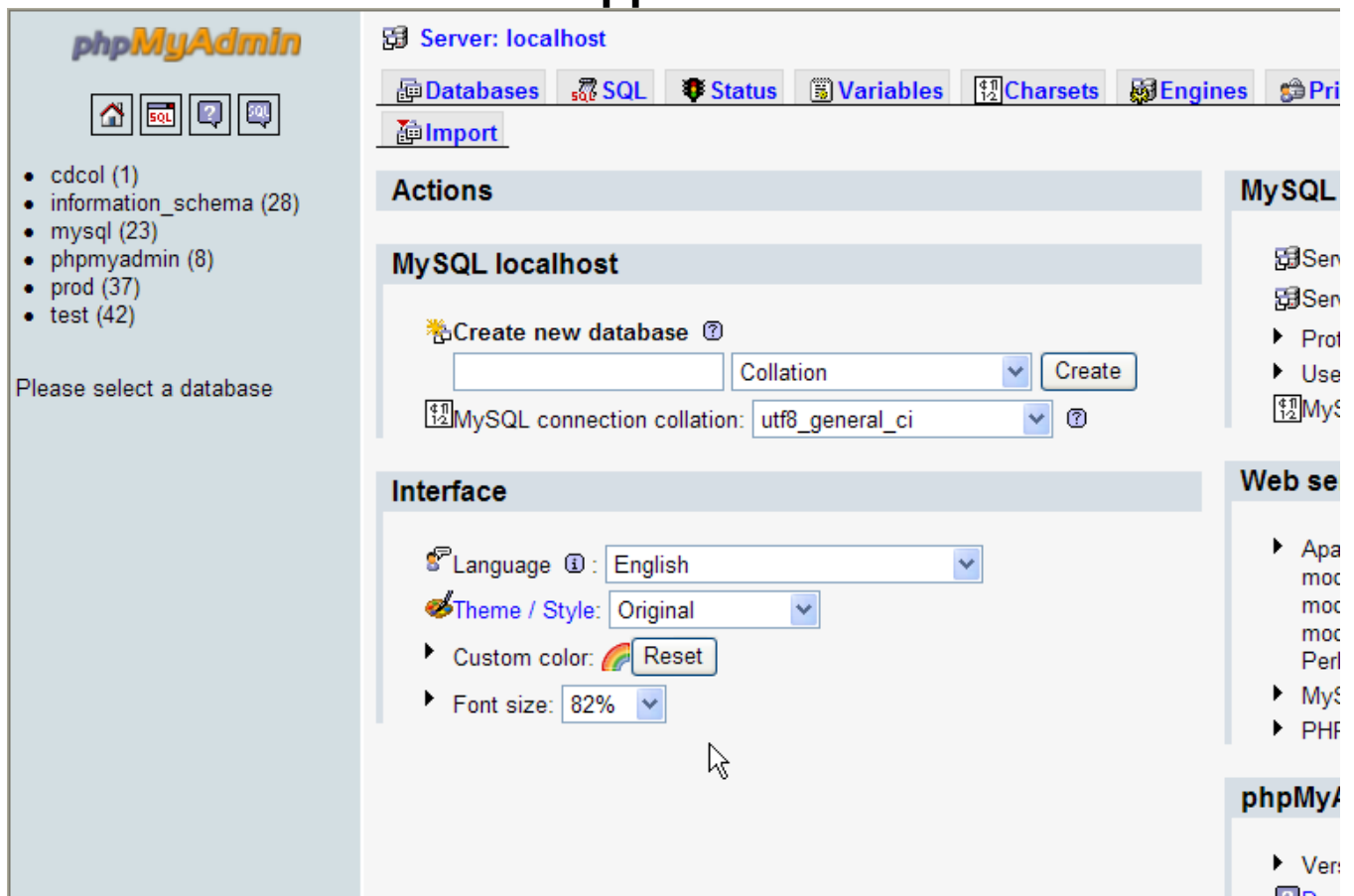
documentation and is relatively easy to use, especially for those who have previous database administration experience.

The menus and forms are fairly user friendly and provide extensive capabilities. This system is accessible using the menu item



Selecting the menu item “PHP Admin” brings up the opening form:

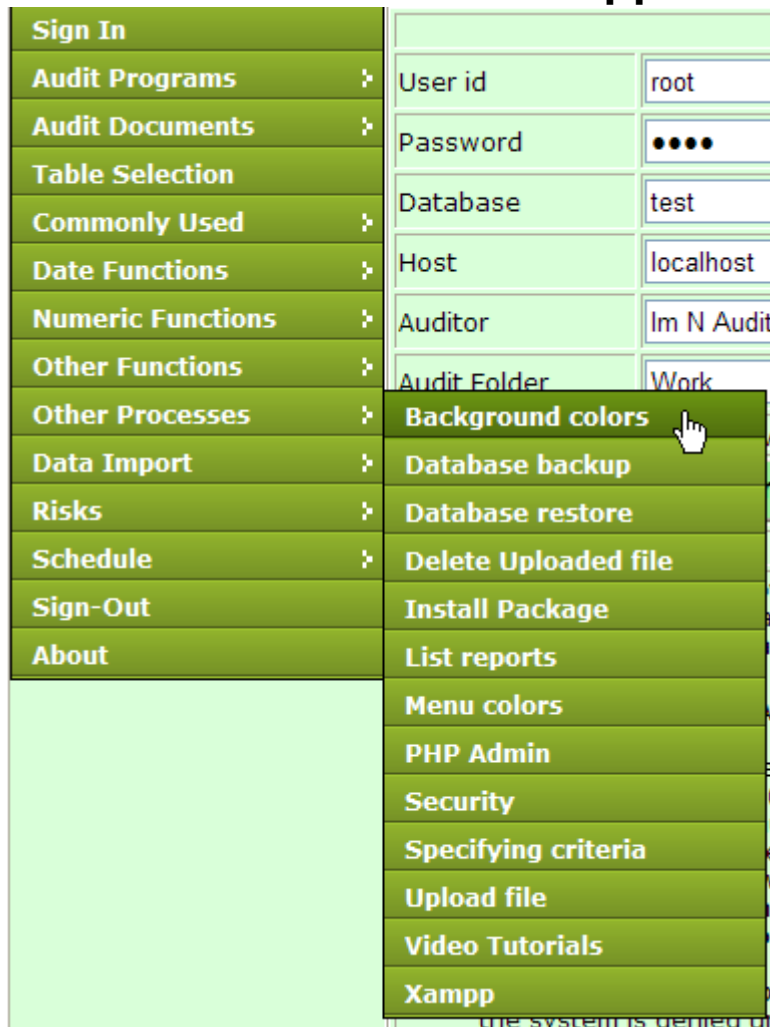
Appendix



6.9 Menu customization

The menus and form background can be customized. Menu colors can be changed using the menu item “Other Processes | Background colors “. The color of the menu itself can be changed using the menu item “Other Processes | Menu color”.

Appendix



6.10 Setting up and scheduling program steps to be run

All scheduling of program steps to be run is done using either “PyCron” (Windows) or “cron” (Unix). Details are provided in section 4.2.

6.11 Installation on Unix internet servers

A prototype system has been installed on a Unix “LAMPP” server. Visit [Web CAAT](#). This is a “live” test system which can be accessed using the id “audit1” and a password of “audit1”. Note that not all capabilities are installed, but this system can provide the “look and feel” of the system as it would be installed on either a local machine or a corporate intranet. It can also be used to see how the system appears using different browsers. Currently (6/17/11) the system is compatible with the most commonly used browsers, but there are some issues with Safari on some of the forms.

Appendix

As the code evolves for the Unix system, it will also be made available under the LGPL license. It is not anticipated that there will be significant differences between the code used for Windows and the code using on a Unix LAMPP server.

6.12 Questions, comments and suggestions

Please provide any comments, suggestions or questions to Mike.Blakley@ezrstats.com (or call me at 919-219-1622). I welcome your comments and will try to respond to all e-mails within **one** business day.