

Basic Techniques for data extraction

A Simplified Procedure

XL Audit Commander

data analysis made easier ...

Basic techniques for data extraction

A frequent audit procedure is to extract data for a specific analysis to be performed. A variety of criteria might be used for the extract, e.g. a specific store number or store number range, transactions under (or over) a certain dollar amount, or where a relationship exists, e.g. book value greater than replacement cost.

The procedures described here are an efficient and effective way to perform both data extractions as well as computations and calculations which are automatically logged and documented. The advantages of automatic logging and documentation are that they reduce the risk of error, as the computations are easier to review.

The procedure requires the use of the XL Audit Commander, a free tool available for download from http://ezrstats.com/Audit_Command.htm. The tool is installed as an Excel add-in, and as such, requires Excel 2000 or later. The tool works only on Windows operating systems.

Techniques for data extraction

Table of Contents

Table of Contents

Basic techniques for data extraction	1
The calculation process (applied to a range of data).....	1
Typical audit areas	1
Test data used.....	1
User Interface.....	1
Procedural Steps.....	2
Data Extraction	3
Computed Values (the calculate command)	5
Related Areas of Interest.....	6
Summary and conclusion.....	8

Procedural steps

Basic techniques for data extraction

The data extraction process may be applied both to data residing in a file, as well as data stored on WorkSheets (or selected ranges thereof). The extraction process takes a data source (such as a file or worksheet) and then applies certain criteria to each row of the data source. When the data row meets the criteria, then it is extracted to a row on another worksheet (previously specified).

The calculation process (applied to a range of data)

The data calculation process is similar to the data extraction process, except that the computed results are stored “in place” for every row in the data source. Thus, a target column will be updated with the results of the calculation. These results may be of various types, such as true/false value, a numeric amount or a text value.

There are several advantages of using a calculation process instead of alternatives such as embedding formula into worksheets:

- The calculations can be expressed in a language that is more understandable (and consequently reviewable)
- The process can be logged (including the formula used)
- This is a repeatable process
- The calculations can be performed as a series of documented steps

Typical audit areas

Practically any area being audited can benefit from data extraction and data calculation procedures, especially where the data being audited exists in electronic format.

Test data used

The test data used in this article is that which is provided with the installation of the XL Audit Commander software. This includes both text files, as well as databases. The worksheets used are those contained in the workbook QS.xls (QuickStart workbook).

User Interface

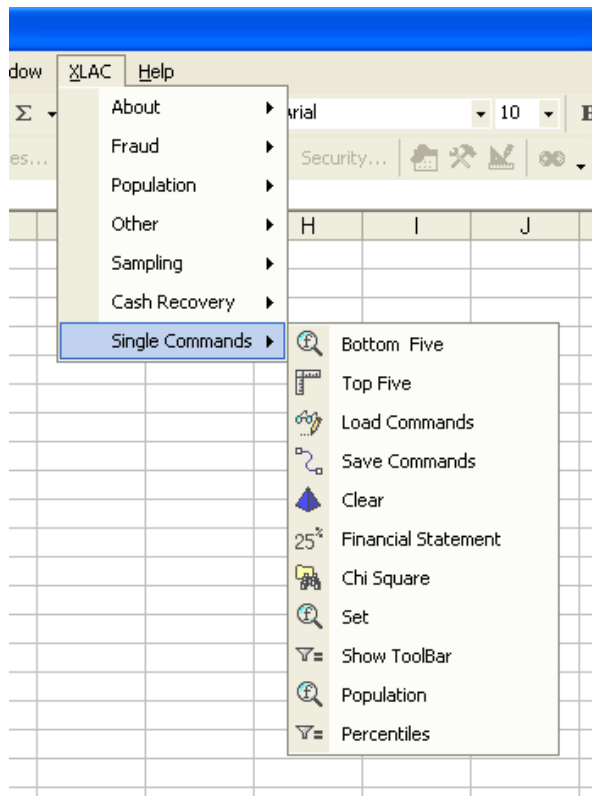
The tool can be used in a combination of four modes:

1. Menu – a graphical system to gather the required information
2. Command Bar – commands are typed as text

Procedural steps

3. System registry – commands are stored in the system registry and then loaded for selection, modification and processing
4. \$Cmd worksheet – the main processing parameters are stored on a worksheet

The menu system allows the necessary data to be gathered in a graphical manner. This menu system is added on to the already familiar menu system of Excel, but with a new menu item labeled “XLAC”. Once the data has been gathered, the “Process” button is clicked and the commands are then written to the toolbar for processing. An example of the main menu is shown below (Note the added menu option “XLAC” which appears at the top of the menu bar).



Commands which can process data from a variety of sources will have a dialog which allows the details of those sources to be specified. After these details have been entered and the “Run Command” button has been clicked, the system will construct the command to be processed and display the command in the toolbar. The command can now be processed or edited.

Procedural Steps

This article focuses on practical areas where the tool can be applied, and provide concrete examples to illustrate each of the commands being discussed. Although there is a menu

Procedural steps

system available, this article will focus primarily on how to construct data extraction criteria and how to compose formulae for calculations.

Both the data extraction as well as the calculations are organized by type of criteria and operators. This is only for the purpose of discussing the expressions. In real world examples, these formula and expressions would be mixed and combine a variety of types of commands, depending upon the specific audit situation encountered.

Data Extraction

Data extraction is performed by specifying three primary items:

1. The data source (file, worksheet, selected range, etc.)
2. The data extraction criteria to be applied
3. The destination for the extract, i.e. which worksheet will be used to store the data extracted

For purposes of this article, the commands have been grouped into the following areas:

- Mathematical operators
- Text operators
- Comparison operators
- Date operators

All of the data extraction commands shown here are included with the installation software for XL Audit Commander (XLACSetup.exe).

Described below are some of the elements of a condition, and their description

Operator	Description	Example
Mathematical Operators		
+	Addition	\$Amount + \$SalesTax
-	Subtraction	\$InvoiceAmount - \$Discount
*	Multiplication	\$Units * \$Quantity
/	Division	\$SalesTax / \$InvoiceAmount
min	Minimum	min(\$InvPaid, \$InvoiceDue)
max	maximum	max(\$DiscAllowed, \$DiscTaken)
abs	Absolute value	abs(\$Discount)
sqr	Square Root	sqr(\$EOQ)
log	Logarithm base 10	log(\$ElapsedTime)
ln	Naperian Log (base E)	ln(\$TimePeriod)
Text Operators		
left	The left portion of the text	left(\$SupplierNumber)

Procedural steps

right	The right portion of the text	right(\$SupplierNumber)
ld	The Levenshtein distance between two pieces of text (used to determine similarity)	ld(\$VendorAddress,\$EmployeeAddress)
Comparison Operators		
<	Less than	\$Amount < 100
>	Greater than	\$Amount > 100
<=	Less than or equal to	\$Amount <= 100
>=	Greater than or equal to	\$Amount >= 100
<>	Not equal to	\$Amount <> 100
=	Equal to	\$Amount = 100
Date Operators		
wd	Weekday 1=Sunday, 2=Monday	wd(\$InvoiceDate) = 6
we	Falls on a weekend	we(\$Invoicedate)
qtr	Quarter number the selected date falls in (can be based on federal, state, calendar or fiscal year)	qtr(\$DatePaid,"F")
ho	date is a federal holiday	ho(\$InvoiceDate)

Each extract command will include a “condition” statement which specifies the criteria which must be met in order for the record to be selected. Note that variable names must begin with a ‘\$’, where the variable name is either contained in the first row of a file, or else the first row of a data range within a worksheet. An example of an extract statement and the results, are shown below.

Objective – extract from range where amount > 100

Command

ex ds=rng sheet=t_CMADData recap=ex2 ulc=a1 cond="\$amount > 100"

Results (log and status bar)

Summary results are reported on both the status bar and the log sheet.

Cmd: EX in: 5,973 items, rpt: ex2 rps: 959 time: 6.226 sec

This message indicates that 5,973 rows were processed, the extract report data is on the sheet “ex2”, elapsed time was 6.226 seconds, or 959 rows per second.

Condition statement	Description of condition
\$amount > 100	Select the row where the amount column exceeds 100

Procedural steps

<code>"Invoice Date" > "8/9/1993"</code>	Invoices after 8/9/1993 – use of quotation marks is required due to embedded spaces and use of forward slashes
<code>\$Col1 > \$Col2</code>	Value in column named Col1 is greater than that in column named Col2
<code>\$Col1 <= \$Col2</code>	Value in Col1 is less than or equal to that in column named Col2
<code>\$Col1 <> \$Col2</code>	Values in the two columns are not equal
<code>\$Col1 = \$Col2</code>	Values in the two columns are equal
<code>Abs(\$Col1) > 100</code>	Absolute value of Col1 is greater than 100, i.e. Col1 is either < -100 or > +100
<code>Log(abs(\$Col1)) > 2</code>	The logarithm, base 10, of the absolute value of Col1 is greater than 2
<code>min(\$Col1,\$Col2) > 100</code>	Either Col1 or Col2 is greater than 100
<code>max(\$Col1,\$Col2) < 50</code>	The largest of Col1 and Col2 is less than 50
<code>left(\$Col1,2) = "AB"</code>	First two characters of Col1 are "AB" (without the quotes)
<code>right(\$Col1,2) = "AB"</code>	Rightmost two characters of Col1 are "AB"
<code>left(\$Col1,2) < right(\$Col2,2)</code>	Left two characters of Col1 are less than the rightmost two characters of Col2
<code>\$date1 < \$date2</code>	Date1 is earlier than date2
<code>\$date1 > \$date2</code>	Date1 is after date2
<code>\$date1 = \$date2</code>	Dates are equal
<code>\$date1 <> \$date2</code>	Unequal dates
<code>wd(\$date1) = 2</code>	Date1 is a Monday (weekday number 2)
<code>Ho(\$date1)</code>	Date1 falls on a Federal holiday
<code>Wd(\$date1) = 7</code>	Date1 is a Saturday
<code>\$date1 < "6/30/2005"</code>	Date1 is prior to June 30, 2005
<code>(\$date1 < "5/31/2005") and (\$date1 > "3/31/2005")</code>	Date1 is between 3/31/2005 and 5/31/2005 exclusive
<code>(\$date1 < "5/31/2005") or (\$date1 > "3/31/2005")</code>	[This condition will always be true, regardless of the date]
<code>(\$date1 < "5/31/2005") or (\$date2 > "3/31/2005")</code>	Either date is prior to 5/31/2005 or date2 is after 3/31/2005

Computed Values (the calculate command)

1. Objective

Procedural steps

Calculate and replace the contents of the column named "Amount" from the file t_CMADData.txt with the contents divided by 2 and store the results on a worksheet named "ca1"

Command

```
ca ds=file file="C:\Program Files\EZS\XLAC\data\t_CMADData.txt"  
col="amount" amount="$amount / 2" recap=ca1
```

Results (log and status bar)

Cmd: CA in: 5,973 items, rpt: ca1 rps: 943 time: 6.337 sec

2. Objective

Same as objective 1 except the data source is a worksheet named t_CMADData

Command

```
ca ds=rng sheet=t_CMADData ulc=a1 col="amount" amount="$amount / 2"
```

Results (log and status bar)

Cmd: CA in: 5,973 items, rpt: t_CMADData rps: 1,656 time: 3.607 sec

Related Areas of Interest

Other related documents/guides of possible interest include:

Topic	Description
Auditing Data in Access	An easier way to perform 18 audit tests on data in Microsoft Access®
http://ezrstats.com/online/AuditGuide/Auditing_Data_in_MS_Access_Databases.pdf	
Auditing Data in Worksheets	18 audit tests for data stored in Excel worksheets
http://ezrstats.com/online/AuditGuide/Auditing_Data_in_Workbooks.pdf	
Auditing Data in Files	18 audit tests to perform on data files in tab separated value format
http://ezrstats.com/online/AuditGuide/Auditing_Data_in_Files.pdf	

Procedural steps

Round Numbers	Why to check for "round" numbers and how
http://ezrstats.com/online/AuditGuide/Testing_For_Round_Numbers.pdf	
Holidays	Identification of holiday dates, e.g. in Journal entries, invoices, etc.
http://ezrstats.com/online/AuditGuide/Testing_For_Holidays.pdf	
Data Stratification	Stratification as a planning and audit tool
http://ezrstats.com/online/AuditGuide/Procedures_For_Data_Stratification.pdf	
Cross tabulations	Use of cross tabulations in audits
http://ezrstats.com/online/AuditGuide/Cross_Tabulations_As_An_Audit_Technique.pdf	
Benford's law	Test conformity with Benford's Law
http://ezrstats.com/doc/Auditors_Guide_to_Tests_using_Benford's_Law.pdf	
Basic Data Extraction	Extracting data based upon criteria, and performing calculations
http://ezrstats.com/online/AuditGuide/Basic_Data_Extraction_Techniques.pdf	
Data Classification	Basic techniques for classifying data Software Installation
http://ezrstats.com/online/AuditGuide/Basic_Data_Classification_Procedures.pdf	
Setup.exe	Setup file - double click to install (6.0 MB)
http://ezrstats.com/online/inno/XLACSetup.exe	
Install Instructions	Installation Guide (PDF document) (.7 MB)
http://ezrstats.com/online/inno/XL_Audit_Commander_Installation_Guide.pdf	
Operation Guide	Operation Guide (PDF document) (2.5 MB)
http://ezrstats.com/online/inno/XL_Audit_Commander.pdf	
Quick Start	Quick Start Module (Excel Workbook - open after install) (3.1 MB)
http://ezrstats.com/online/inno/QS.xls	
Help	Shows list of help links in the current workbook
http://ezrstats.com/helpxlac/he.php	
Single Commands	Commands of just two letters for a selected range on a single worksheet
http://ezrstats.com/helpxlac/single.php	
Population	Population statistics (univariate, stratify, population, duplicates)
http://ezrstats.com/helpxlac/ndxpop.php	
Sampling	Sampling procedures (cma, interval, sample size calculation, precision calculation)
http://ezrstats.com/helpxlac/ndxsamp.php	
Fraud	Fraud investigation tools (test Benford's Law, duplicates)
http://ezrstats.com/helpxlac/ndxfraud.php	
Cash Recovery	Cash Recovery procedures ("Near miss" invoices, split invoices)
http://ezrstats.com/helpxlac/ndxcr.php	

Procedural steps

Other	Other Commands (ageing, gaps, credit card validation, analytic review procedures, dates on federal holidays, etc.)
http://ezrstats.com/helpxlac/ndxoth.php	

Summary and conclusion

Data extraction and calculations are an essential part of almost every audit which involves audit data stored in electronic format. There are many techniques for performing data extraction and making calculations. The advantages of the approaches described include the following:

- The calculations can be expressed in a language that is more understandable (and consequently reviewable)
- The process can be logged (including the formula used)
- This is a repeatable process
- The calculations can be performed as a series of documented steps
- No macro or other programming language required

It is hoped that auditors can employ these techniques in order to perform audits in a more efficient and effective manner.