

# Quick Reference Sheet

## ACL Command Reference

Test or Operation	Explanation	Command(s)	Key fields*/Records	Output
Basic	Typically the commands a data analyst would use when first working with a data file. They provide general information about the file or about a subset of the data when applied in conjunction with a command filter.	<i>Verify</i>	C N D	Screen, File
		<i>Count</i>	Record-based	Screen
		<i>Total</i>	N	Screen
		<i>Statistics</i>	N D	Screen, File
Completeness	Allows the user to verify that all records in a sequence are present, e.g., check numbers.	<i>Gaps</i>	C N D	Screen, File
Uniqueness	Can determine whether or not records have been included more than once. It can also determine whether or not a particular field contains unique values.	<i>Duplicates</i>	C N D	Screen, File
Frequency & Materiality Distribution	Can determine how many records and how much value is concentrated by time period, value range, or other record identifiers such as location codes, vendor/customer numbers, or product identifiers. These commands are also useful for identifying outliers, either in financial or age terms.	<i>Stratify</i>	N	Screen/Graph, File
		<i>Classify</i>	C	Screen/Graph, File
		<i>Age</i>	D	Screen/Graph, File
		<i>Summarize</i>	C D	Screen, File
		<i>Cross Tabulate</i>	C	Screen/Graph, File
Multi-File Combinations, Comparisons, and Associations	Associate or combine data from separate files.	<i>Extract/Append</i>	Record- and field- based	File
		<i>Join</i>	C N D	File
		<i>Relations</i>	C N D	
Sampling	Conduct record-based and monetary unit samples.	<i>Sample</i>	Record- and numeric field-based.	File
Benford's Law	Tests leading digits for variance from standard distribution. May indicate fraud.	<i>Benford</i>	N	Screen/Graph, File
Order	Refers to testing the records for logical order. This test is useful as it may show that it is unnecessary to sort or index.	<i>Sequence</i>	C N D	Screen, File
Re-Ordering	Allows the user to change the order of the data in the current file or in a new file containing the same data.	<i>Sort</i>	C N D	File
		<i>Index</i>	C N D	Screen

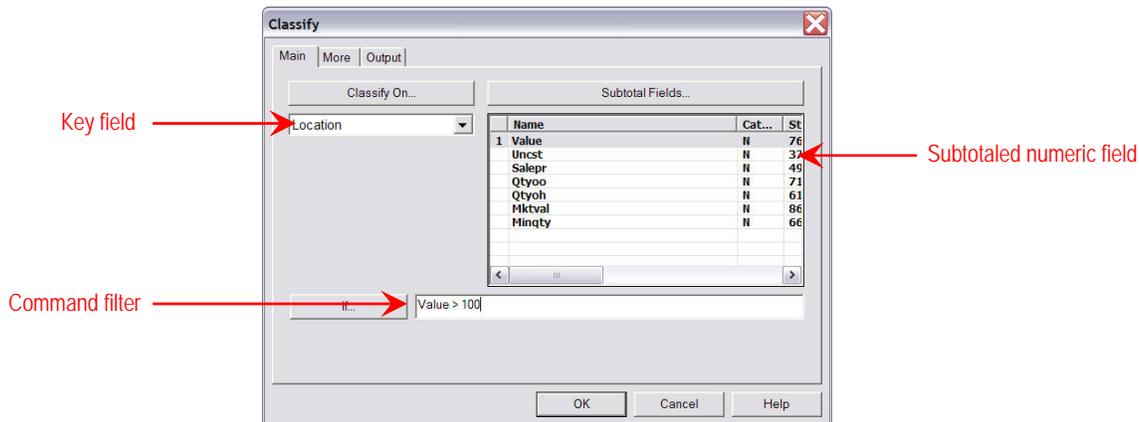
\*C=Character N=Numeric D=Date

# Quick Reference Sheet

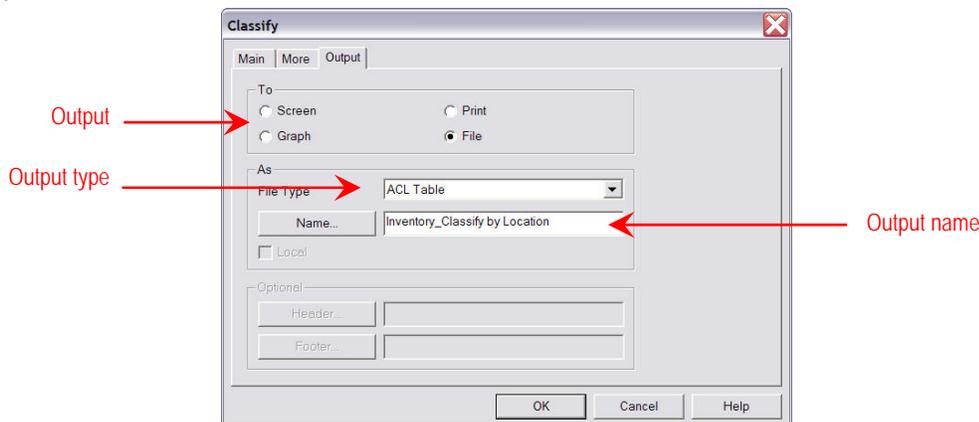
## ACL Command Terminology

Term	Explanation
Parameters	Any user-defined or user-selected criteria that govern how the command is executed. Examples of parameters: <ul style="list-style-type: none"> <li>▪ Key fields</li> <li>▪ Subtotaled numeric fields</li> <li>▪ Additional field information</li> <li>▪ Command filters</li> <li>▪ Output type/name</li> <li>▪ Limits on how many records will be processed (More tab)</li> </ul>
Key field(s)	The field(s) against which the command is run. Certain commands can only be run against specific field types, either character, numeric, or date.
Command filter	A filter local to the command that is active only while the command is running. Once the command has completed execution, the filter becomes inactive.
More tab	Governs the number of records (scope) that will be processed. <p><b>First:</b> Allows you to test the command against the first x records. This can save time if you have a very large file and you would like to test the command against a limited number of records.</p> <p><b>Next:</b> If your record indicator is not set at the first record, this will allow you to process the next x records.</p> <p><b>While:</b> Executes the command only when a certain condition is met, either a record-based condition or a variable-based condition. This parameter is useful primarily in scripts.</p>

### The Main Tab



### The Output Tab



# Quick Reference Sheet

## Basic Commands: VERIFY, COUNT, TOTAL, & STATISTICS

These ACL commands on the Analyze and Data menus provide general information about the data file regarding field definitions that are appropriate to the data contained in the field, possible data corruption, missing or invalid dates, record counts, numeric totals, bounds, and statistical information on numeric and date fields. It is a best practice to run VERIFY and STATISTICS against numeric and date fields before beginning analytical procedures and comparing the results to control totals and other file information supplied by your data provider.

Command	Toolbar Button	Key field(s)	Comments
VERIFY		Numeric, date, character	Compares field type to data contents. Identifies invalid or blank dates and numeric fields containing non-numeric characters. Useful for verifying that field definitions are appropriate.
COUNT		Record-based	Confirms number of records in table and documents the result in the log. Generates <i>COUNT1</i> variable.
TOTAL		Numeric	Confirms control totals for numeric fields in the table and documents the results in the log. Generates <i>TOTAL1</i> variable.
STATISTICS		Numeric, date	Confirms record counts, control totals (numeric), date bounds, negative and positive numeric values, means, and bounds. Documents results in the log. Generates multiple variables (see list below).

### Explanation of Output for STATISTICS (Numeric field)

Item	# of records	Materiality	Average for item	Item description: <i>VARIABLE(S)</i>
<b>Range</b>	-	110,967.60	-	→ Difference between maximum ( <i>MAX1</i> ) and minimum ( <i>MIN1</i> ) values: <i>RANGE1</i>
<b>Positive</b>	145	694,361.94	4,788.70	→ Records with positive values in this field
<b>Negative</b>	5	-13,882.00	-2,776.40	→ Records with negative values in this field
<b>Zeros</b>	2	-	-	→ Records with values of zero in this field
<b>Totals</b>	152	680,479.94	4,476.84	→ All records: <i>COUNT1, TOTAL1, AVERAGE1</i>
<b>Abs Value</b>	-	708,243.94	-	→ Absolute value of total materiality in field: <i>ABS1</i>
<b>Std. Dev.</b>	-	9,784.72	-	→ Measure of dispersion around the average (optional): <i>STD1</i>

	Highest	Lowest	
MAX1 →	100,800.00	-10,167.60	← MIN1
	37,100.00	-2,774.40	
	25,548.60	-595.20	
	24,738.00	-190.72	
HIGH1 →	23,136.00	-154.08	← LOW1

## Quick Reference Sheet

### Basic Commands: VERIFY, COUNT, TOTAL, & STATISTICS

#### Explanation of Output for STATISTICS (Date field)

Most of the output of the STATISTICS command when it is run against a date field is irrelevant. The real value of such a procedure is to identify the time period within which the transactions occurred. This can be determined by examining the oldest (lowest) and most recent (highest) transaction dates. This is a quick way of confirming that the data provided meets your specifications with regard to cutoff.

	Highest	Lowest	
Most recent →	04/20/2007	05/01/2006	← Oldest
	04/20/2007	05/01/2006	
	04/20/2007	05/01/2006	
	04/20/2007	05/01/2006	
	04/12/2007	05/01/2006	

#### STATISTICS Shortcuts

To isolate the **top ten transactions by value**, run STATISTICS on the numeric field with the number of high/low parameter set to 10. Then run the EXTRACT command with the command filter *name of value field>=HIGH1*

To isolate the **ten oldest transactions**, run STATISTICS on the transaction date field with the number of high/low parameter set to 10. Then run the EXTRACT command with the command filter *name of transaction date field>=HIGH1*

#### Identifying Outliers with STATISTICS

Outliers are transactions that occur outside of reasonable expectations. These expectations are usually, but not always, based on the transaction's materiality. Other types of outliers can be based on the age of the transaction, the volume (units), unit value (price), or classification codes such as vendors, product numbers/classes, or geographic regions. Outliers can also be based on combinations of multiple parameters, such as a very large transaction in a geographic region where the average transaction is relatively small.

**Numeric fields:** Run STATISTICS to get an overall picture of the bounds of that field. Then run STRATIFY to see the number and materiality of the transactions based on size ranges. Drill down from the screen output to isolate the outliers with a view filter. The *standard deviation* (see below) can also provide helpful information regarding outliers when comparing two populations.

**Date fields:** Run STATISTICS to get an overall picture of the period covered by the transaction data. Then run AGE to see the number and materiality of the transactions based on age ranges. Drill down from the screen output to isolate the outliers with a view filter.

For outlier detection with other ACL commands, see *Commands: Frequency and Materiality Distributions*.

#### What is Standard Deviation?

The standard deviation of a population is a measure of dispersal of the values around the average (mean). It is useful when you are comparing statistics from two different populations and it can help identify which population is more likely to have outliers.

For example, if you are comparing transaction data from two locations, and one location carries high-value products while the other carries a mix of high-, medium-, and low-value products, it is possible that both would have the same average transaction size but different standard deviations because of the product mix.

In a normally distributed population, 68% of the values would be  $\pm 1$  standard deviation from the mean, 95% within 2 standard deviations, and 99.7% within 3 standard deviations.

In the example on the previous page, this would mean that 68% of the values would lie between  $-5307.88$  ( $4476.84 - 9784.72$ ) and  $14261.56$  ( $4476.84 + 9784.72$ ).

## Quick Reference Sheet

### Commands: Frequency & Materiality Distributions

These ACL commands on the Analyze menu are useful for identifying concentrations of transactions or materiality by specific field characteristics. When analyzing all your data, they can provide breakdowns of the entire population. After you have isolated exceptions or possible indicators of fraud, these commands can provide common characteristics of the suspect transactions that could indicate the reason for the control failure or the source(s) of the fraud. These commands are also useful for identifying outliers. All of these commands can subtotal multiple numeric fields. When the output is directed to the screen, you can see it either as text in the example below or as a graph by selecting the "Graph" button at the bottom left-hand corner of the view.

Command	Toolbar Button	Key field(s)	Comments
CLASSIFY		Character fields, one at a time.	<b>Comparing Two Populations:</b> After isolating exceptions or suspicious transactions, you can compare the relative frequency or materiality by running Classify against the entire population and against the exceptions. Then use RELATIONS on the key field to bring the two results into one view for comparison.
SUMMARIZE		Multiple character/date fields.	
STRATIFY		Numeric fields, one at a time.	<b>Financial Auditors:</b> Subtotal numeric fields for materiality.
CROSS-TABULATE		Multiple character fields; creates pivot-table output.	<b>Operational Auditors:</b> The Count column (see example below) will tell you how many transactions meet each criteria.
AGE		Cutoff date fields.	

### Explanation of Output for CLASSIFY, STRATIFY, & AGE

Key field	# of records (frequency)	% of frequency	% of materiality	Subtotalled numeric field (materiality)
Location	Count	Percent of Count	Percent of Field	Value
01	18	11.84%	5.24%	35,661.98
02	54	35.53%	39.53%	268,988.06
03	37	24.34%	29.39%	199,966.67
04	22	14.47%	12.31%	83,753.57
05	13	8.55%	6.24%	42,479.36
06	7	4.61%	7%	47,609.10
22	1	0.66%	0.3%	2,021.20
Totals	152	100%	100%	680,479.94

Possible outlier as location code does not match the pattern of other codes and contains only one transaction.

## Quick Reference Sheet

### Commands: Frequency & Materiality Distributions

#### Comparing CLASSIFY and SUMMARIZE

Functional Specification	Classify	Summarize
Calculates and reports the number of times a key field appears in the table	Yes	Yes
Computes and displays subtotals on selected numeric fields	Yes	Yes
Computes and displays percentages	Yes	No
Key field can be character	Yes	Yes
Key field can be date	No	Yes
Number of key fields allowed	One	One or more
Allows the display of other fields	No	Yes
Data needs to be sorted on key field prior to running command	No	Yes, presort option available
Primary location utilized to handle sequence process, subtotalling of numeric fields, and compression of details	RAM	Hard disk
Can send results to screen	Yes	Yes
Can send results to table	Yes	Yes
Can send results to graph	Yes	No

#### Identifying Outliers

Outliers are transactions that occur outside of reasonable expectations. These expectations are usually, but not always, based on the transaction's materiality. Other types of outliers can be based on the age of the transaction, the volume (units), unit value (price), or classification codes such as vendors, product numbers/classes, or geographic regions. Outliers can also be based on combinations of multiple parameters, such as a very large transaction in a geographic region where the average transaction is relatively small.

**Numeric fields:** Run STATISTICS to get an overall picture of the bounds of that field. Then run STRATIFY to see the number and materiality of the transactions based on size ranges. Drill down from the screen output to isolate the outliers with a view filter.

**Date fields:** Run STATISTICS to get an overall picture of the period covered by the transaction data. Then run AGE to see the number and materiality of the transactions based on age ranges. Drill down from the screen output to isolate the outliers with a view filter.

**Character fields:** Run CLASSIFY against the specific field to identify the number and materiality of the transactions based on that field. If there are any codes that are not recognizable, drill down from the screen output to isolate the transactions associated with those codes.

**Multiple fields:** For character and date combinations, use SUMMARIZE. For character field combinations, use SUMMARIZE or CROSS-TABULATE. If you want to use a command that is designed for character fields against a date or numeric field, use conversion functions such as STRING() or DATE() to create computed character fields. You can also create filters with multiple conditions to identify outliers with specific characteristics.

## Quick Reference Sheet

### Commands: Multi-File Combinations, Comparisons, and Associations

Command	Toolbar button	Key field(s)	Output
EXTRACT		Record- or field-based.	New ACL table containing the specific records/fields that have been extracted and appended from the original tables.
JOIN		CND	New ACL table containing the matched or unmatched records based on the user's parameter selection.
RELATIONS		CND	None. Fields from related tables can be used in computed fields, filters, and commands in the parent table.

#### EXTRACT

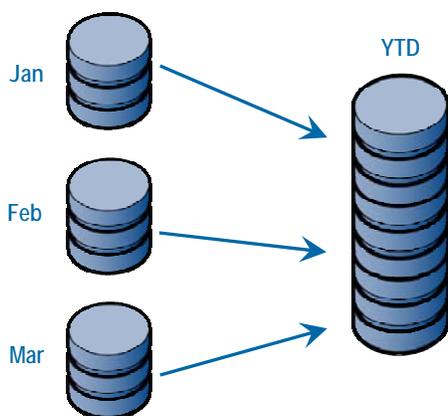
The Extract command allows you to isolate specific records or to combine records from multiple tables into one master table.

##### Isolating Specific Records

Once you have determined that exceptions or suspicious transactions exist in your data, you can extract those records to a separate table for further analysis to determine what may have caused them. Analyzing the exceptions in a separate table is more efficient than analyzing them in their source file as the exceptions table will be much smaller. You can isolate them by running Extract with a command filter where the filter criteria match the criteria for non-compliant transactions. For example, if all purchase orders over \$5000 require VP approval, you would run Extract with the following command filter: *amount>5000 AND ISBLANK(Approval)*.

##### Combining Records

You may often receive data from multiple time periods or from multiple locations that you wish to combine into one master file. You can do this by selecting the Append option on the More tab when running the Extract command from each source table to the master table. In the example below, you would combine the January, February, and March transactions to create the YTD table.



**Best Practice:** Always Extract/Append to a new table rather than to one of the source tables.

**Record-based Extract/Append:** Ensure that table layouts are identical in the source file and the destination file. The fields must have the same length and the same physical order.

**Field-based Extract/Append:** Ensure that the selected fields have the same length in both files and that they are selected in the same order as they exist physically in the destination file.

## Quick Reference Sheet

### Commands: Multi-File Combinations, Comparisons, and Associations

#### JOIN

The Join command allows you to compare two tables so that the result can yield either matching or unmatched records based on the user's parameter selection. The two tables must have at least one field in common, and that common field must be of the same type (CND) and length in both tables. The output consists of a new table containing all the requested records from either the primary or secondary table, or both depending on which type of Join is selected.

In the examples below, we will be comparing a payroll file to an active employee master file for different types of tests.

#### Example 1: Matched Join

Your objective is to verify that the gross pay of each employee payroll transaction matches the gross pay in the active employee master table for all valid employees. A valid employee is defined as one that exists in the master table. In this case, you would run a Matched Join with the employee number as the key field in the Primary (payroll) and Secondary (master) tables; select the employee number as the key field in each table and select the gross pay fields as output fields along with the employee number. The output file will contain the payroll transaction for each employee that exists in the master file, and you can use a filter to identify any employees for whom the gross pay does not match up.

#### Example 2: Unmatched Join

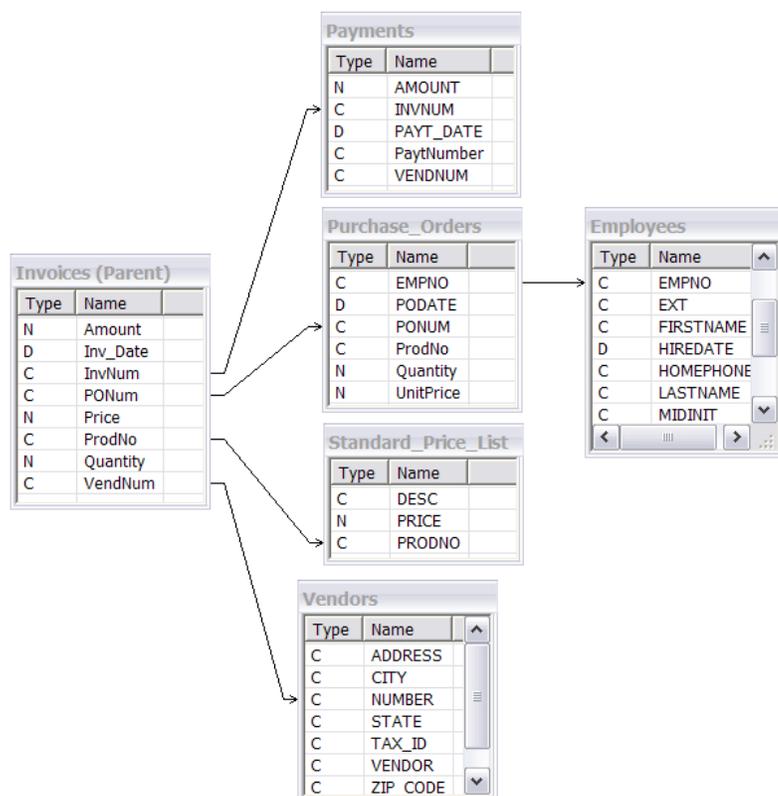
Your objective is to identify payroll transactions that were generated for employees who are not in the active master employee table; they have either been terminated or their employee number is not valid. In this case, you would run an Unmatched Primary Join with the payroll table again as the Primary, selecting all primary fields for output. The resulting table will contain all payroll transactions for invalid/inactive employee numbers.

All Primary/All Secondary Parameter: This forces the output of all primary table records or all Secondary table records in a Matched Join. This is useful if you want to keep one of the tables whole for the purpose of examining the materiality or extent of the exceptions in relation to the entire universe of records.

The first five Join options on the More tab are many-to-One in nature. This means that if there are any duplicate instances of the key field in the Secondary table (such as the employee number), the first match encountered in the Secondary table will be used and the second will be ignored. The sixth Join is many-to-many in nature and all possible matches will be written to the output file.

#### RELATIONS

Relations links multiple tables to a parent table (usually a transaction table) for multiple lookup capability. This allows enhanced, sophisticated cross-table testing and analysis.



In the purchasing example at left, all the other tables have been directly related to the Invoices table with the exception of the Employees table, which is indirectly related via the employee number field in the Purchase Orders table.

These relations will allow the following tests to be executed in the Invoices table:

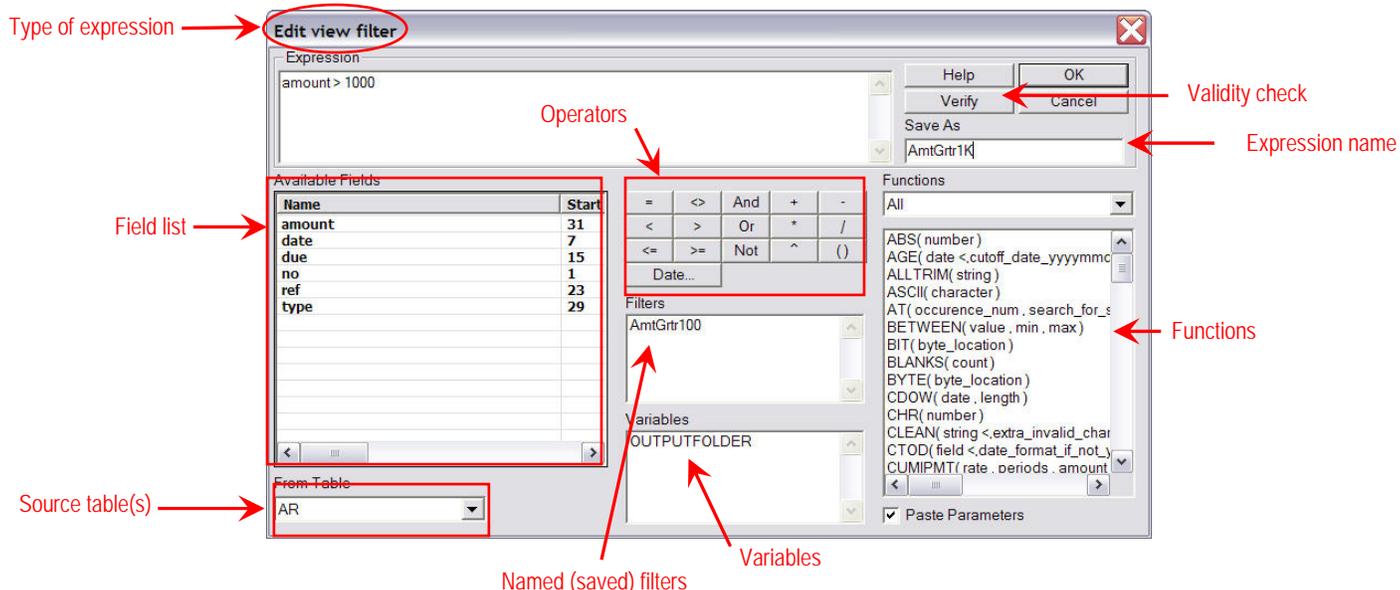
- Identify invoiced prices variance from standard price and calculate amount of variance
- Identify invoices with unmatched products and quantities compared to purchase orders and calculate variance
- Identify vendor/employee/product profiles of non-compliant transactions and materiality
- Compare vendor-employee addresses, phone numbers, and other identifiers
- Calculate days between purchase order and invoice
- Compare all invoices to master vendor list to identify inactive/invalid vendors

# Quick Reference Sheet

## ACL Expressions Reference: The Expression Builder

### EXPRESSION BUILDER

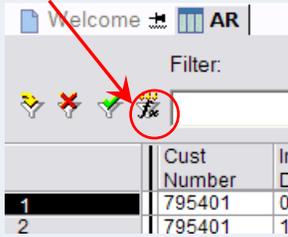
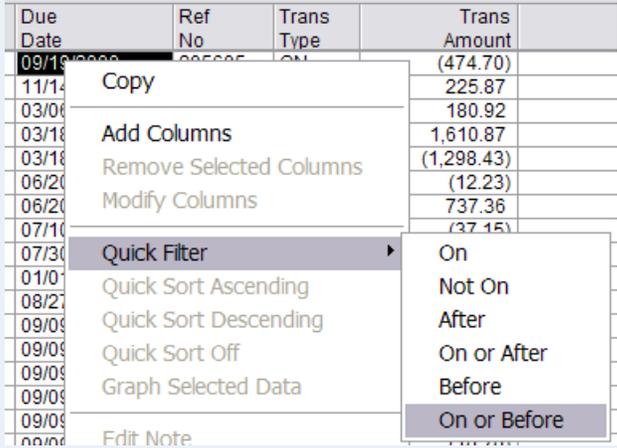
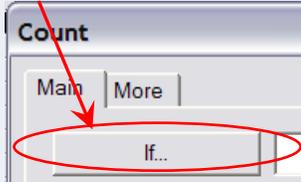
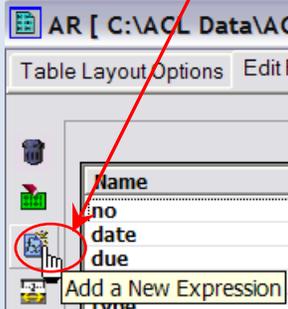
The expression builder is used for creating filters and computed fields using fields, operators, variables, functions, and named filters.



Element	Explanation	Best practice/tips
Expression name	Usually used for naming filters to make them permanent	Use a meaningful name such as <i>AmtGrtr100</i> . Avoid using <i>test</i> , <i>junk</i> , or <i>temp</i> .
Field list	All fields, physical and computed, from the source table	Click on the column headings to sort the field list. This will help you find fields faster. You can sort in reverse order as well. Sort by category to find specific field types.
Functions	Allow you to create more powerful and complex expressions	Three sources of information on functions and their application: Online Help, Online Learning Functions module, and ACL 301 instructor-led training course.
Named filters	All named filters from the source table	You can edit, copy, and delete named filters by going to Edit>>Filters.
Operators	Logical operators	Use brackets to organize your expressions for more precision.
Source tables(s)	The currently opened table or any related tables	Use Relations to access data in other tables for comparison with the current table or for computations based on fields from multiple tables.
Type of expression	Indicates if you are creating a view filter, command filter, or computed field	
Validity check	Click Verify to test the expression's validity	If you are building a complex expression, such as a filter with multiple tests, click Verify after building each test. It will tell you if the expression is valid up to that point and allow you to identify any logical failures.
Variables	System and user-defined variables	To make a variable permanent, provide it with a name that begins with an underscore (_). You can edit variables by going to Edit>>Variables.

# Quick Reference Sheet

## ACL Expressions Reference: Filters and Computed Fields

Types of expressions	Categories	Creation point
<p><b>FILTERS</b></p> <p>Search/query for records that meet the filter criteria. Can be named to save permanently to the current table layout. Commands executed while a filter is active will only process the records that meet the filter's criteria. Filters can be based on more than one criterion, and can refer to fields of all data types within the current table and in any related tables.</p> <p><b>Result data type:</b> L (logical)</p> <p>Applications: Identify exceptions, possible indicators of fraud, and outliers. Drill-down into current table through graph output from Classify, Stratify, and Age. Document number of exceptions in the log by running Count while the filter is active. Isolate filtered records in a separate table by using Extract with command filter.</p> <p><b>Syntax:</b> The minimum required syntax for any filter is field-operator-string/value.</p> <p>Character: LastName = "Smith"</p> <p>Numeric: Amount &gt; 100000</p> <p>Date: TransDate &lt;= '20060101'</p> <p>Note the required formatting for each data type. You can combine, group or exclude multiple criteria by using AND, OR, NOT and ().</p> <p><b>Important:</b> If you are working with a very large number of records (e.g., more than 100,000) you may get a time-out message when imposing a view filter on the data. This type of filter is known as a thin filter as it is looking for a very small number of records in a very large table. To avoid time-outs, you can:</p> <p>Extract the desired records to a separate table with a command filter attached after turning the view filter off.</p> <p>Re-set the Redraw Seconds option to a higher number in the View tab in Tools&gt;&gt;Options. This is not considered a best practice as it is less efficient.</p>	<p><i>View</i></p> <p>All commands will be executed against filtered records when a view filter is active.</p>	<p>Edit view filter button. Opens the Expression Builder.</p> 
	<p><i>Quick Filter</i></p> <p>Based on cell values in the view.</p>	<p>Select and right-click cell(s) in view. Resulting view filter can be edited in the Expression Builder.</p> 
	<p><i>Command</i></p> <p>Executes command against filtered records, after which filter is inactivated.</p> <p>Extract the desired records to a separate table with a command filter attached after turning the view filter off.</p> <p>Re-set the Redraw Seconds option to a higher number in the View tab in Tools&gt;&gt;Options. This is not considered a best practice as it is less efficient.</p>	<p>If...button in command dialog. Opens the Expression Builder.</p> 
<p><b>COMPUTED FIELDS</b></p> <p>Virtual data based on a formula. Added permanently to table layout when named. Can be based on fields within the current table and in any related tables.</p> <p><b>Result data types:</b> C, N, D</p> <p><b>Commands:</b> Commands can be run against computed fields in the same way that they can be run against physical data fields.</p>	<p><i>Unconditional</i></p>	<p>Edit table layout &gt;&gt; Add a New Expression</p> 
	<p><i>Conditional</i></p>	

# Quick Reference Sheet

## ACL Expressions Reference: Computed Fields

### Unconditional Computed Fields

Apply the same formula to all records.

*Field name:*  
Recommend c\_ at beginning to indicate that it is computed.

*Green check mark:* Click to save computed field.

Click to open expression builder. Final formula appears at right.

*Format:* Governs how numeric field will be presented in the view.

*Alternate Column Title:* Column heading that appears in the view and in reports.

### Conditional Computed Fields

Apply different formulas to different records based on conditions (filters).

*Default Value:* Result if none of the conditions are met. Best practice: Use "Unknown" for character fields to trap instances of possible unrecognized codes.

*Condition/Value pairs:* Different result for each condition. ACL tests each record beginning with the first condition; if that condition is not met, it proceeds to the second. The order of the conditions is thus important.

*Insert a new condition:* Create new condition/value pair using the Expression Builder.

Condition	Value
Location = "01"	"Paris"
Location = "02"	"Berlin"
Location = "03"	"London"
Location = "04"	"Hong Kong"
Location = "05"	"Tokyo"
Location = "06"	"New York"

# Quick Reference Sheet

## ACL Expressions Reference: Functions

### FUNCTIONS

Functions are prewritten instructions that perform computations on specified data. Most functions require one or more input parameters. Use them to add power to your filters and computed fields and derive more value from your data.

Category	Examples
Conversion	Convert numeric to character and vice-versa.
Date	Identify day of week, convert to character and vice-versa, automatically use today's date.
Logical	Wild-card search, search for multiple matches, isolate range of records, test data pattern for consistency, test for blank entries.
Math	Absolute value, rounding, modulus, selecting largest/smallest values of a pair.
String	Data harmonization, isolating components within one field.
Other	Tag records with record number for tracking, compare values between records (up or down).

### Computed Field Example

Many legacy systems contain name information in one field rather than discrete fields for first, middle, and last names. The challenges are (1) the individual name segments do not always occupy the same bytes in each record; and (2) names may or may not have middle names or initials.

Using the `SPLIT()` function in computed fields, you can parse the name field into its components based on the number of blanks that separate the first, middle and last names within each record as identified by the `OCCURS()` function:

Name	Last	First	Middle
Andrew Jacobs	Jacobs	Andrew	
Myrtle Eunice O'Donnell	O'Donnell	Myrtle	Eunice

Last Name: `SPLIT(Name, " ", OCCURS(ALLTRIM(Name), " ") + 1)`

First Name: `SPLIT(Name, " ", 1)`

Middle Name: Default value: `SPLIT(Name, " ", OCCURS(ALLTRIM(Name), " "))`

Condition: `OCCURS(ALLTRIM(Name), " ") = 1`

Value: ""

### Filter Example

Input validation is the process by which a system tests user inputs for appropriateness before being accepted by a system. Legacy systems tend to have weak input validation, and additional ad hoc testing is often necessary to validate user inputs. One example is an identification number that should follow a consistent pattern and format, such as a social insurance number or a social security number. The `MAP()` function can be used to test for content and format and will identify any records where that field is not consistent with the requirements.

In this example, the required format is Alpha-Number-Slash-Number-Number, such as k3/72. The filter should search for the records where the field does not match the format:

ID Number	Filter Result
a1/22	F
b518	T
33/y6	T
z4/93	F

The filter expression is written as follows:

`NOT(MAP(ID_Number, "x9/99"))`

where x is the universal indicator for an alphabetic character and 9 is the indicator for a numeric character.

Run the Count command with this expression as a command filter to document the existence of any exceptions. Run the Extract command with the same command filter to isolate non-compliant records in a separate table.