

7 habits of highly successful auditors?



Best Practices for Naming Conventions in Projects

Submitted by Porter Broyles, President ACL Houston User Group, ACL Super User

What do the Big Four and world class internal audit organizations have in common? They both have high expectations clearly defined in their audit standards to promote consistency and professionalism. Audit standards ensure audits are performed, preserved, and presented in a professional manner. Yet, many organizations have done little or nothing to standardize their CAATTs function. Senior management usually does not know or understand the CAATTs function and leaves it up to their experts to establish standards. As a result, the quality of these standards varies greatly from company to company.

In many cases where there is a lack of standards, the deficiency becomes problematic when a CAATTs practitioner tries to read another user's project. On small projects, this deficiency may not be a problem. However, on larger projects, with scores of scripts and tables, interpreting another users' process might be impossible —

- Which tables contain the original data, or represented final deliverables?
- Which fields were included with the original data vs. fields users created?
- Which scripts call which subscripts?
- How do you read the project?
- Where does one begin?

Proper documentation, both inside and outside of the project, is crucial. Many companies have the minimum CAATT standards - recording the original data sources and detailed explanations on what was done - but fail to establish one of the most important standards: naming conventions. Good naming conventions help make ACL projects easier to interpret by both internal and external reviewers, saving time and effort. It helps the reviewer decipher what the programmer intended.

When working on an ACL Project there is four areas to consider for best practices in naming conventions:

- 1) Fields
- 2) Variables
- 3) Scripts
- 4) Tables

1) Fields

Most ACL users create descriptive terms when naming new fields to help remember the field contents. Many projects are thus filled with field names such as "Account", "Business Unit", "Amount", or other common terms. Unfortunately, database administrators often use descriptive terms when naming new fields for the same reason. This means that many databases contain field names such as "Account," "Business Unit," "Amount" or other common terms.

This poses two significant problems. First, it becomes very difficult, if not impossible to determine which fields are user created and which were in the original data sources. Second, it poses the problem that a standardized script may define a field using the same name as the original data. This is particularly a concern when the script overwrites or deletes the field before recreating a new field with the same name.

As a best practice, user-defined fields should be readily identifiable. One method is the inclusion of a prefix. For example, rather than naming a field "Amount", use the name "F_Amount." The prefix, "F_", identifies the "f_" field as a user created field. Anyone reviewing the script instantly knows that fields with the prefix "F_" are user-defined. Likewise, any field without a prefix is an original piece of data.

2) Variables

Variables are identifiers that represent different values. They can be used to represent different pieces of data in a Script. Again, best practice is to use variable names that clearly identify the variable. One method is to add a prefix to variables such as "v_". This is a generic identifier for variables. This convention allows users to quickly convert variables to field names. For example:

```
DEFINE FIELD F_amount COMPUTED v_amount  
Here the v_ variable is used to create a f_ield.
```

While "V_" is the standard prefix for variables, it does not have to be the only one. Other potential variables include (but are not limited to):

```
VL_ or L_ for Index names  
VT_ or T_ for Table name variables  
VD_ or D_ for Date variables  
VL_ or L_ for List variables
```

For both fields and variables, the key isn't what prefixes are used, but rather that they are used consistently and that everyone in the department understands their usage.

3) Scripts

Often when reviewing Projects, it is impossible to tell which Scripts are subscripts to which main Scripts or the order in which they are executed. Ideally Scripts should be named in a manner that creates an outline in the project navigator. Best practice is to have the first scripts run at the top of the navigator and the last ones run at the bottom like a table of contents in a book. To this end, an Alpha-Numeric system can be utilized. The basic Alpha-Numeric system uses a Letter followed by two numbers (i.e. A01, D04, J02).

Using this method, the first Script can be named "A00_Script_Master". This places the Script at the top of the project navigator. It is the script that controls all other subscripts within the Project. Often it is necessary to solicit information, define fields, and/or normalize data. The Scripts that are run before the analysis Scripts could be labeled A01-A99 followed by a short description (i.e. A01_Identify_tables, A02_Select_fields, A03_Normalize Data, etc.)

Another best practice is to have a 'clean-up' Script at the end of the Project. This Script, 'Z00_Clean_up' can be written to delete temporary files and unnecessary user computed fields. This is important especially if the Script is to be repeated on a regular basis with new data added to older data.

Between the A00 and Z00 Scripts you can have the analytical scripts. In more complex scripts, each major analytical step should have its own ALPHA character (i.e. looking for duplicate addresses, unusual accounts, invoice gaps, etc.). Each of these analytical approaches might have its own ALPHA character. Duplicate addresses might be D00; unusual accounts might be E00; Invoice gaps might be F00.

Under each of the major analytical steps, there might be a number of subscripts. These can then be labeled x01-x99. If a subscript requires a subscript, they can include additional alpha-characters after the initial numeric (i.e. x01a x02b.). Subscripts should start with Alpha-Numeric code of the script that calls them. This maintains the table of contents and makes it obvious which Scripts are utilized (i.e. D01a will be a subscript of D01. D01a01 will be a subscript of D01a which is a subscript of D01). This makes it easy to figure out the logic and flow of a Project.

4) Tables

Naming conventions surrounding tables are fairly basic; there are four types of tables in every Project.

First, tables that contain data retrieved directly from the system should be recognizable as such. Anyone who reviews the Project should be able to recognize which fields contain original data. One method is to add a prefix or suffix that identifies the table. Possible conventions include "OD_" for Original Data, MASTER_, Original_, AAA_, or even the company's name/initials.

Second, tables may contain data derived from the original tables, but are utilized repeatedly during the execution of the scripts. For example, two tables with original data may be merged together to create a working table. Again, these tables should be readily identifiable as secondary sources.

Third, there are the tables that serve as intermediaries between the working tables and the final results. These are the files can be deleted at the end of the script or subscript. Many users will use the prefix TEMP_ followed by a short descriptor to identify temporary files. A better practice, however, uses the table names "Temp1" through "TEMPXX." This convention provides two advantages: by numbering them, the order in which they were created becomes readily apparent and it makes it easier to troubleshoot problems; and it's easy to create a generic clean-up script that works with every script you might use.

Finally, there are the final products or report tables that contain the results of a Script. A good practice for identifying report files is: REPORT_(Alpha-Numeric)_(descriptor). The Alpha-Numeric portion relates the Script naming convention to be described above. The descriptor is a short identifier indicating what the report contains. For example, the name might be "REPORT_D01_Duplicate_Search". Using this convention it is easy to identify results, what they are, and what produced them.

Like all standards and naming conventions, those utilized by an organization are up to the people who work there. Standardization supports the preservation of the organization's investment in CAATs tools and services and saves time and effort.

■ acl.com
info@acl.com

ACL, the ACL logo, the ACL logo with the text "Data you can trust. Results you can see.", and Audit Command Language are trademarks or registered trademarks of ACL Services Ltd. All other trademarks are the property of their respective owners.