Paper 094-29

Balancing Data Quality Against Time and Money Constraints

Tricks from the Trenches

Susan B. Long, Syracuse University, Syracuse, NY
Linda Roberge, Syracuse University, Syracuse, NY
Jeffrey T. Lamicela, Syracuse University, Syracuse, NY
Mummoorthy Murugesan, Syracuse University, Syracuse, NY

ABSTRACT

Data quality is vital. Decisions made and actions taken based on incorrect data have the potential for disaster. While you can't afford NOT to clean your data, you never have the necessary resources to ensure that your data are truly clean. Therein lies the paradox. How do you balance these two competing needs – ensuring data quality but within a limited time frame and on a limited budget? At the Transactional Records Access Clearinghouse (TRAC), a research center at Syracuse University, we have been engaged in the quality-versus-resources struggle since 1989. We hope that the practical lessons we've learned along the way and the SAS tools we've developed in the process can be used as a guide for others.

INTRODUCTION

Concern about the quality of data is nothing new. For as long as we have been collecting and using data, we have been concerned with whether or not they have been recorded and processed properly. While the technology for recording information has evolved from the advent of punch cards to today's distributed processing systems, problems with data quality seem to be ever more intractable. In a 1980 paper on managing research data, Ronald Helms described his plan for handling the "irregularities [that] will inevitably occur" due to data collection problems and as a result of merging data from numerous sources.(1) He specifically addressed how data "purification" must be accomplished in the face of time, budget, and computer processing constraints.

While this problem isn't new, it has neither been solved nor lessened in severity. Helms was concerned with research data, and specifically excluded data collected for administration such as personnel records, accounting data, inventories, and other types of transactions. While Helms' view in 1980 was that administrative data was less complicated and thus less likely to be erroneous, an additional twenty plus years of experience suggest otherwise. The IT literature abounds with war stories about time wasted, money lost, bad decisions, etc. all resulting from erroneous data. (e.g., 2, 3, 4, 5) Poor data quality has even been implicated in serious scientific disasters. (6)

Two developments have served to exacerbate the data quality problem. First, the amount of administrative data that are captured and stored has increased. Databases have increased in number, in size, and in the richness of detail that is captured. It is rare to find even small businesses that don't have at least a portion of their processes automated and information stored in relational databases. Entries in our old rolodex files have ballooned from name, address, and phone number to include cell phone, fax, pager, email address, URL and any other information that may be available. The growth in data means that we can expect a concomitant increase in wrong data (e.g. a wrong phone number entered), outdated information (e.g. mailing address not updated), and numerous inconsistencies that make the data less usable.

Second, advancements in technology such as inexpensive storage and fast processors that have fostered the development of administrative data have also produced wonderful new applications. The use of data warehousing, in particular, has exploded. So too has the development of decision support tools to allow managers to query and analyze information in real time, and to use data in new ways to help inform them on an ever expanding list of strategic policy concerns.

These new capabilities also place greater demands upon the quality of our data. We routinely merge data from multiple sources including legacy systems and administrative applications. Do merged datasets use the same data representations and categories? New composite data have provided rich sources of information used for everything from social program evaluation to corporate decision support to oversight of government programs. (7, 8, 9) The result has been that data developed for use in one application are often used in multiple applications that may be far from the purpose for which they were originally intended. Do these newer users have the same understanding of the meaning of the data that the original developers did?

Luckily, as our awareness of data quality problems has increased, our view of data cleaning has changed. Once seen as an unimportant task, it is now understood as an essential function. (10) Investigation into the causes and cures for "dirty data"

has become an important area of concern for data managers. (11, 12)

However, most organizations still approach data quality on a piecemeal basis, with a primary focus upon fixing "defects" after they occur – hence the terms data cleaning, data scrubbing, data cleansing which tend to be used to encompass desirable data quality efforts. This stands in stark contrast to quality control (QC) in other business arenas – where the focus of QC is to prevent defects from occurring in the first place, rather than in repairing defects after they occur. The QC approach to data quality views the production of information as a process, which needs to be continuously monitored and adjusted to ensure that the resulting "product" emerges as defect-free as possible.

TRAC AND THE TRACFED DATA WAREHOUSE

One's perspective on data quality is forged in part by one's professional training, and in part by one's experiences in the data trenches. Professionally, while the authors come from somewhat diverse backgrounds, we are all members of the Transactional Records Access Clearinghouse (TRAC) at Syracuse University. Since TRAC forms the backdrop that has helped shape our joint perspective, we thought a brief description of the Center would help place our suggestions into perspective.

TRAC is a nonprofit, nonpartisan research data center associated with Syracuse University, with offices on campus and in Washington, D.C. The center is self-supporting, and is primarily grant funded with user fees offsetting only a fraction of our costs. Unfortunately, rarely do funders recognize the day-to-day costs of ensuring quality data. And thus, while data quality efforts are at the core of what we must find a way of doing, we have never been able to find funders willing to foot the bill for such efforts as an express project goal.

TRAC has designed, built, and runs a data warehouse composed of administrative and other types of data. Since 1989, TRAC has used the Freedom of Information Act to gather a variety of transactional data from various agencies and departments within the federal government. We then used these combined data sources to build and continually update large knowledge bases which track federal staffing, expenditures and day-to-day enforcement efforts. These databases are used to build our data warehouse and to carry out specific research projects. Both the resulting information and reports are then made available to the public, most recently via the World Wide Web. Currrently, TRAC's data is available on two web sites: a public or free site (http://trac.syr.edu, see Figure 1) and a subscription site (http://tracfed.syr.edu, see Figure 2).

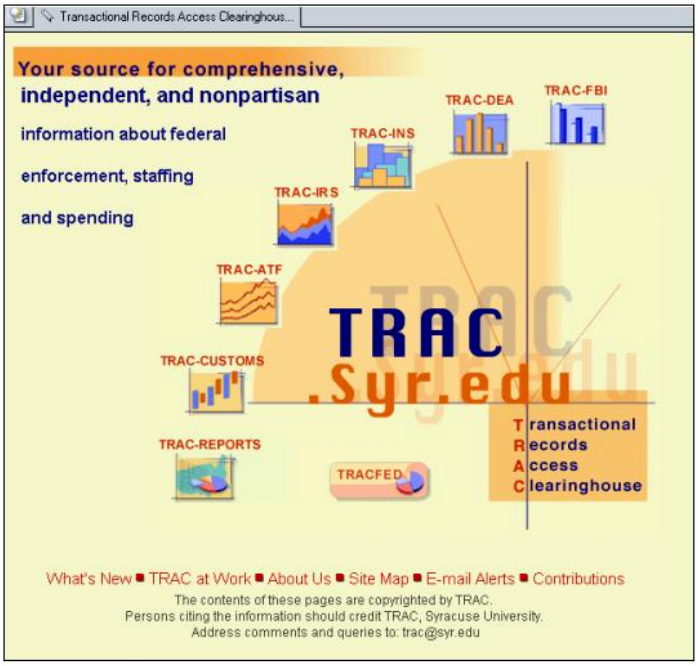


Figure 1: TRAC's Free Website

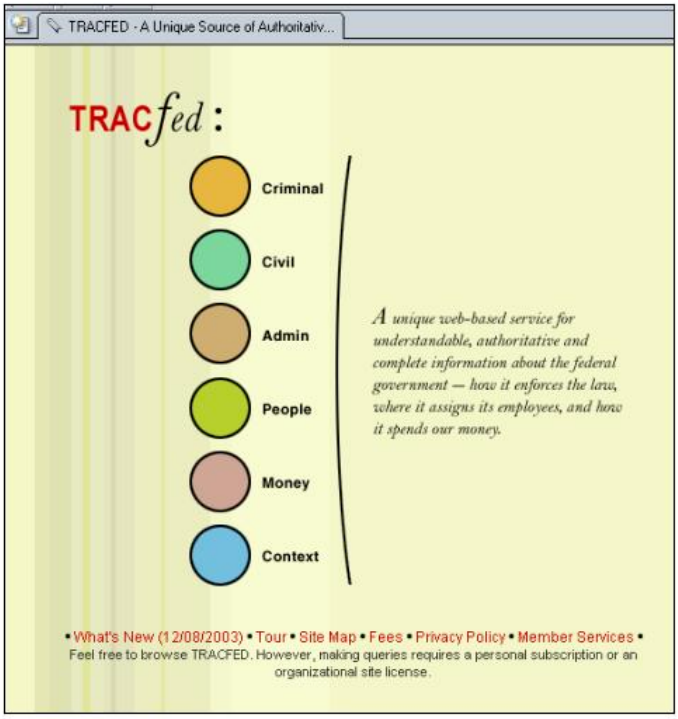


Figure 2: TRAC's Subscription Website

The public site provides information in the form of thousands of static web pages that are produced by analyzing data in the data warehouse. This site offers information about various federal government departments such as FBI (Federal Bureau of Investigation), DHS (Department of Homeland Security), DEA (Drug Enforcement Agency), IRS (Internal Revenue Service), ATF (Alcohol, Tobacco, and Firearms), as well as special topics such as criminal terrorism enforcement since the 9/11 attacks. There are limited abilities for users to tailor the information to suit their purposes.

The subscription site provides dynamic access to the wide range of data in the data warehouse via specially designed data mining tools.[1] These tools allow users to focus their explorations, compare their findings to specifically chosen benchmarks, discover trends, and drill down to more detailed levels. Users[2] also have the ability to create unique subsets of the data, which they can store on our servers in their own "Web Lockers" and use for further analysis.

For TRAC, data quality is vital. Not only does our reputation rest on it, but our core purpose – providing useful information – would be undermined if people could not rely upon our data and findings. But as any organization we face practical constraints – both in time and resources. First, our informational holdings are not insignificant. With nearly a terabyte of online storage, our databases consist of millions of records and thousands of different variables. We continually bump up against the frustration of having more data than we have either the time or staff to adequately assess.

---

[1] There are six different types of data that make up TRAC's data warehouse:

- criminal workload of the Department of Justice;
- civil actions where the U.S. is a party including causes of action relating to civil rights, employment, freedom of information and privacy, antitrust, consumer protection, frauds, torts, forfeitures, taxes, prisoner suits, and bankruptcy;
- administrative enforcement related to IRS audit and collection activities;
- federal civilian workforce including U.S. district court judges and prosecutors;
- distribution of federal funds; and
- community context.

Data mining tools use SAS/IntrNet® product to produce tables, listings, graphic displays and data maps.

[2] Professionals from many different backgrounds use the TRACFED data warehouse for a wide variety of purposes. TRAC's users come from criminal justice, business, public policy, political science, journalism, public administration, and law. Organizations having site licenses include libraries, the media, public interest groups, government agencies, and Congress. As we discuss later, the variety of purposes for which our data are used makes it difficult to balance quality with resources.

Second, we have no control over the originating organizations that produce the data. This means that we are severely limited in our ability to correct problems at their source. In addition, government agencies, while the source of our data, are not typically willing partners in our efforts, and frequently resist providing us their data. Sometimes we have to resort to lawsuits to force the release of information we need.[3] Consequently, government officials often take an adversarial stance and don't necessarily share information needed to help us monitor data quality, or alert us when their definitions and internal practices change. The burden falls entirely upon us to ferret out such problems.

Finally, because our users and their objectives are so varied, we can't restrict our attention to a few key information fields. In short, we face many of the same problems that face other data warehousing efforts.

APPROACHING DATA QUALITY SYSTEMATICALLY

In the interests of full disclosure and at the risk of disappointing everyone even before we begin, we want to preface our remarks by emphasizing that we have no magic answers. And we offer no simple solutions. All we can offer is to share some of the strategies and approaches we have adopted. We have singled out five of these strategies to discuss here.

1. KEEP A PROPER PERSPECTIVE

It is important to recognize that, when it comes to data quality, zero-defects is rarely a reasonable goal. Failure to accept this fact from the onset is almost sure to get you into trouble. As an illustration, consider this personal story on how one of the authors painfully learned this lesson.

> *As a principal investigator on a research project, a newly-minted PhD at Princeton University set out to re-use data from a leading research study on criminal recidivism to test some additional research questions. As a first step in this research, she loaded the data into a SAS data set and began a quick review the entries in each field to see if codes conformed to the codebook and if the information contained was internally consistent. The data set followed experimental subjects for two years, and provided detailed chronologies on their employment, their family and housing status, and their arrests and incarcerations. She quickly discovered myriad problems – including major inconsistencies. For example, persons were recorded as gainfully employed (an indicator of treatment success) at the same time they were locked up in prison (an indicator of treatment failure). So a clean-up effort was launched. But even after months of effort, no end was in sight. Most discouraging, the results seemed little influenced by the cleanup effort.*

Many of you may have had similar experiences. Attacking dirty data can be a seemingly bottomless pit. Thus, keeping a proper perspective is essential. Managing a data project requires constant weighing of the costs of potential errors against the costs of seeking their removal. Finding the middle ground – somewhere between really dirty data and zero defects – is never easy.

The better you understand how the data will be used, the easier it will be to determine the trade-offs. For example, in medical records, failure to record an immunization may not have much impact if there is no harm in giving a second dose. On the other hand, noting that an immunization has been given when, in fact, it was not can have much more serious repercussions. Failure to note an allergy to penicillin can be deadly when the information is used to determine treatment for an infection but of little consequence when used as ancillary information for a study on weight management.

2. WHAT YOU CAN'T SEE *CAN* HURT YOU

Often the problems you can see are not the only or even the most important problems. Too many data quality initiatives zoom in on individual records and define data quality in terms of ridding each record of incorrect entries. Less frequently do data quality efforts focus attention on whether the correct set of records made it into the database. Our experience is that in terms of impact, problems of coverage often dwarf other data quality concerns. Let us share a horror story to illustrate this – but it is just one of many we have encountered.

> *One of the federal agencies TRAC followed early on was the Nuclear Regulatory Commission. Among its duties, the NRC is responsible for monitoring the safety of commercial nuclear power plants. We obtained of several of their databases, including the one that tracked every inspection made at a plant along with the results of that inspection – Were safety violations found? Were any actions taken? Information from this database was used widely not only within the NRC itself, but reports were regularly released by NRC summarizing the safety record of each of the power companies*

[3] While the Freedom of Information Act requires public access to information about federal agency activities, not all agencies cooperate. Litigation may then be required. Currently we have three active lawsuits seeking the continued release of data needed to update our current data series.

that operated these plants. These summaries were relied upon by others, including Wall Street firms rating bonds issued by energy companies on the risks and likely returns.

At a certain point in time we noticed that the volume of recorded safety violations by the NRC had fallen sharply. Checking further, we found that the number of records (inspections) contained in file had also dropped off at the same time. We contacted the NRC to find out whether we had received complete files for the recent time periods. We had. We then asked NRC if they were only doing 20 percent of the number of inspections they had done in the past. That got their attention since the number of inspections had not in fact fallen. As it turned out, the NRC was transitioning to a new database system, and running the two systems in parallel to facilitate a smooth changeover. During this transition, forms had to be separately input into both systems. But no increase in staff time was provided. Changeover schedules had slipped so the transition kept being postponed. Checking turned up stacks of inspection forms in many of the regional offices that had not been input into either system. No one had noticed that their database was incomplete, or that all of their published reports generated from these files were incorrect. To sort out the mess, the NRC had to hire an outside contractor to check their paper records and reconstruct their database going back for several years.

At TRAC, we have found that aggregate comparisons need to be part of our regular data monitoring system. If you are lucky, you may be working with a database where you know just how many records to expect. For example, if there is one record for each product line for each office in the organization. "Are all records present?" is then an obvious question to ask.

Often, however, it is not easy to know just how many records complete coverage would represent. Here we find multiple checking strategies essential.

If over time you repeatedly process the same database, having a system which retains aggregate counts for past time periods to compare with the newly processed file can be informative. Retaining some aggregate breakdowns – by relevant offices or types of cases or both – can enhance these comparisons. We find longer time periods so that you have many counts in the series help you spot discrepancies and separate these from normal variation. [We discuss under topic 5 below the system we have developed that does this.] Any sudden drops (or jumps) require an explanation. More often than not in our experience, sudden change in volume is a data recording issue rather than a sudden change in actual events.

Wherever possible, also follow pending workload and make sure counts balance. Pending cases at the beginning of a reporting period should equal pending cases at the end of the prior reporting period. Numbers pending at the beginning of a period, plus new receipts should equal the numbers pending at the end of the period plus matters closed. It is surprising how often this is simply not the case in published annual reports, at least for governmental organizations.

Where feasible, a third approach is to compare record counts across more than one database. This works if you have overlapping data systems tracking the same set of events. For example, output from one office becomes the input for another; are the volumes roughly the same? Federal criminal court cases are tracked by Administrative Office of the U.S. Courts, by the U.S. Department of Justice that is the plaintiff in criminal actions, and by the federal investigative agency that conducted the investigation. How do the number of income tax prosecutions, for example, compare in each system? Because definitions of how and what events get tracked can vary by data system, comparisons across systems pose some challenges and cannot be expected to be exact, but order of magnitude and consistency in differences can still be informative.

We want to emphasize that here we are not doing record-by-record matches. We are simply comparing aggregate counts. While record-by-record matches have great value (see below), they usually are resource intensive on several levels, and are not always feasible if there is no common unique key.

What we suggest here is to compare counts after records are aggregated. This is usually quick and easy to do and can help pinpoint where you may have a data quality problem that requires more in-depth investigation.

3. THE CASE OF THE DISAPPEARING RECORDS

Good database systems impose stringent controls upon how records, once added, can be removed. But we still encounter situations that surprise us. We are currently working trying to unravel the source or sources of one such surprise. We call it the case of the disappearing (occasionally reappearing) records. Situations like this call for record-by record matches to be incorporated as a matter of routine.

The U.S. Attorney office in each federal judicial district maintains an Oracle database system that tracks its workload. Each matter recommending that someone should be prosecuted that is received from law enforcement agencies is separately tracked in this database, and followed until the matter is declined or if the defendant is prosecuted and the court proceedings are concluded. Each month, a script is run in each of the 93 offices to produce an extract of new and updated records, which then is then transmitted to the national headquarters and used to update a national file.

Records are removed from the national file when cases have been formally closed out and actions concluded, or after a certain period of months when no updates have been received (when they move to a "delete history file"). Nonetheless, we found when we matched records in the national file for adjacent months, that a number of records simply disappear from the file and never appear on the delete history file. The volume of records that disappeared – overall about 7 per cent, rising in some subcategories to one-third or more – was significant. Occasionally a record, after being missing for several months, would suddenly reappear in the national file, but usually it simply disappeared without a trace.

How could this happen? At this point we simply don't know, but it is clearly a data quality problem that needs to be pursued. Not only do the disappearing records have the potential to alter findings but, until an answer is found, raise the potential spectre of a database system that may be hiding other deficiencies because it is not under proper controls.

In some situations, even a single "lost record" can have horrific consequences. Some of you may recall a situation a few years back in which the child protective services agency in Florida lost track of a child the agency was responsible for protecting, with had tragic consequences for the child.

While feasibility and cost considerations must be weighed, we now lean towards incorporating exact record matches into all of our routine processing procedures. In the case we cite of U.S. Attorney records, we had not originally done so. But we were alerted to the possible problem by our aggregate checks (point 2 above). We now have added a record-by-record match to our monthly processing updates, and carry along a record even if it has disappeared from the latest national file.

One element in our match routine may be worth mentioning. We have added a new field to each record to record the outcome of our monthly match. Its width is the number of months we have made the comparison (which repeats on a twelve month cycle). A '1' is entered when the record was present that month and a '0' when it isn't. Thus, a '111110000000' indicates that the record was present for the first five months and then dropped out for the remaining 7 months of the year, while a '000011110000' indicates a record first appeared in month 5, was present for four months, and then disappeared. Using these distinctive patterns we can quickly identify subsets of records for further investigation, monitor dropped records over time, as well as test out alternative theories against how often records disappear for different types of cases.

4. TRUST, BUT VERIFY

Never assume that what "isn't possible" doesn't find a way of occurring. It is easy to be lulled into making assumptions about your data. Just because it never has been a problem in the past, don't assume it can't become a problem at some point in the future. Try to build in quiet checks in your SAS code to alert you if these built-in assumptions are violated. (What we call a "quiet" check results in no extra output *unless* a problem is detected.)

Building in checks is of course a cardinal rule for all programmers. But it is sometimes easy to overlook that your code is predicated upon a data-specific assumption, particularly when your data has had a uniform history of always being "this way." Reading through your code to look for these hidden data assumptions at each step can save you headaches down the road.

Unfortunately, this is a lesson we continually relearn by getting caught unaware ourselves. Common areas to look for are:

- At a merge, use a by statement and always check for mismatches even though you think they couldn't possibly occur.
- In a conditional processing step which depends upon predefined data values, build in a prior step to ensure that nothing in those values has unexpectedly changed, reflecting some unannounced coding change or modification in error checking routines so that something new has slipped through.
- Never write code that, in order to work properly, assumes the data set you are processing only has the variables defined in prior steps, and never those defined in subsequent steps. Otherwise you can get some really unexpected results if you find it necessary to rerun part of a processing sequence.
- If your processing assumes that a required field always has an entry, and it always has in the past, check first to make sure no missing values unexpectedly crop up at some future point. Even if such checks are unnecessary ninety-nine point nine percent of the time, they may save your bacon when a problem does occur.

And never assume that your own processing of the data doesn't add in something unexpected. That's why we can't say enough for PROC COMPARE in SAS. We use it religiously whenever we touch our master data sets. This includes the use of any procedure that rewrites the data set, even if it shouldn't alter any real content, or the correction is a minor one that updates a value in one or a few records out of thousands or even millions. It is insufficient to verify that the intended change took place. You need to also verify that something else wasn't screwed up. In one step, PROC COMPARE can verify that there was indeed no change where none was intended, and that any intended change occurred precisely where it was wanted.

5. THE CHEAPEST WAY TO IMPROVE DATA QUALITY

Perhaps the surest way to improve data quality is to build tools to allow users to access the information. Neither organizations nor individuals have much incentive to invest time in seeing that data are accurate if they don't get any benefit (or suffer any harm) from it. Good data cost money. Considerable time and effort is required, as well as persistent monitoring. To repay making this investment, sufficient benefit must accrue – and the incentives must be tied as directly as possible to those charged with its creation. The most direct incentive is of course the value of having information at your fingertips when you need it.

Further, as a rule, correcting data defects after they occur is usually much more costly than exerting additional care to prevent their occurrence in the first place.

This is why we firmly believe that the cheapest way to improve data quality is to encourage its use. This is true even in our situation, where we have little control or influence over the government agencies that input the data we use. A recent example of the impact of TRACFED on data recording practices at the Department of Justice amply illustrates how increasing data usage can create feedback that in turn causes improvements in data quality.

> *The U.S. Department of Justice for more than several decades has classified its activities under various program categories. Usage of this data had until recently been fairly limited. TRAC took this information and used it in its data warehouse, making the information accessible on the World Wide Web. There anyone, including an enterprising reporter from the Philadelphia Inquirer covering the government's war on terrorism after 9/11, was able to produce a listing of each and every criminal prosecution brought that was labeled as a "terrorism" case. Further investigation uncovered that many of the cases appeared to have little connection with terrorism at all. (13) The publicity that this news coverage engendered reached Congress and resulted in the General Accounting Office being asked to look into the matter. The GAO concluded that "better management oversight and internal controls [were] needed to ensure accuracy of terrorism-related statistics." (14) Needless to say, greater care is given by the agency on how this data is now entered.*

While not all examples of how increased data usage lead to improvements in data quality are quite this dramatic, time and experience have taught us that building usage leads to improvements in data quality.

Providing access through an intranet or internet site is one of the most effective approaches for increasing data use. We have found the SAS/IntrNet® product makes providing access in this manner to SAS data sets fairly quick and easy.

We have found one of the tools built for our external users to be particularly helpful in-house as we assess data quality and look for error problems while processing our data. This is our drill down, or "going deeper" data mining tool. This tool lets you examine data from different perspectives, and easily go from aggregate counts down to sets of similar records or further down to display the detailed fields for an individual record. It is this combination of viewing perspectives and the ease with which you can go from one perspective to another that we find particularly useful in uncovering data anomalies and then diagnosing their source.

Details of this tool, including some relevant code, is included in Appendix A. As our starting point, we used the example code for drill down applications which SAS has distributed on its web site. (15) The data needed for this application are both a file with your individual records and a summary output from PROC SUMMARY (or parallel procedure) which aggregates these detail records according to one or more classification variables. For our criminal database, for example, we summarize by time, geography, investigative agency, lead charge and program category. This allows easy comparison over time, location, type of case, etc. which allows one to spot data anomalies more easily.

We have added a number of additional features to the original code provided by SAS. These include the ability to drill down to a listing of the individual records that contributed to the statistic. Then if additional detail is desired beyond the listing of records, each individual record on the list can be selected and a pop-up window then display all of the values contained in the fields pertaining to that observation (even across several interrelated data files). Automatically built-in hyperlinks on the detail page can lead you even further to display related cases.

We have been pleasantly surprised at how much more efficient our search for data anomalies has become, and we frequently build a such a drill down application just to allow us to more efficiently survey new data acquisitions to help us decide whether the data is of sufficient quality to add to our data warehouse. Because the SAS code is written in a generic manner, once you create a summary data set from a detailed one, little more is required than to revise the libname location for your data and provide a list of your classification variables used in summarizing your data along with the variables you want to display, and you are in business.

CONCLUSION

While it is important to try to correct data defects through cleaning techniques, because of limited funds and time to devote to data quality, we have also found it crucial to look at information production as a process to be continually monitored and adjusted. By working to carefully weigh costs against benefits, compare aggregates, detect missing records, build in "quiet checks", and develop tools which make the data easier to use, we have managed to maintain and improve the quality of our data even as its volume grows. This has resulted in improvements in the processes used by the organizations that produce the information, as well increased value of the information to those who have come to rely on its integrity.

CONTACT INFORMATION

We encourage readers to visit our websites and to contact us with any comments or questions. For more information and a free trial subscription, SUGI participants can go to http://tracfed.syr.edu/sugi.html. All authors can be reached at:

Transactional Records Access Clearinghouse
488 Newhouse II
Syracuse University
Syracuse, NY 13244
Voice: (315) 443-3563
Fax: (315) 443-3196

E-mail addresses:
Susan Long -- suelong@syr.edu
Linda Roberge – lroberge@syr.edu
Jeffrey Lamicela – jlamicel@syr.edu
Mummoorthy Murugesan - mmuruges@syr.edu

REFERENCES

1 Helms, RW. (1980), "A Distributed Flat File Strategy for Managing Research Data." *Communications of the ACM*, 80, p279.

2 Greengard, S. "Don't Let Dirty Data Derail You." (November, 1998) *Workforce*.

3 ____ "Bad Data Management Costing Companies $1 Billion." (May 7, 2001*) Accounting Today* v15, i8, p16.

4 Nelson, K. "Bad Data Plagues ERP." (March, 2002) *Bank Systems and Technology,* v39 i3 p12.

5 Nguye, T. (2003) "The Value of ETL and Data Quality." *Proceedings of the 28th Annual SUGI Conference*, SAS Institute, 161-28.

6 Fisher, CW and Kingma, BR. "Criticality of Data Quality as Exemplified in Two Disasters." (2001*) Information and Management*, 39, p109.

7 George, RM and Lee, BJ. "Matching and Cleaning Administrative Data." (June, 2002) *New Zealand Economic Papers*, 36, p63.

8 Ballou, DP and Tayi, GK. "Enhancing Data Quality in Data Warehouse Environments." (January, 1999*) Communications of the ACM*, 42:1, p73.

9 Burnham, DB and Long, SB. "Judging the Judges" (December, 2002) *Champion: The National Association of Criminal Defense Lawyers*, p12.

10 Brauer, B. "Data Quality – Spinning Straw into Gold". *Proceedings of the 26th Annual SUGI Conference*, SAS Institute,

117-26.

11 Hanks, ML. "Cleansing Looms Important in Data Warehouse Efforts." (February, 1999). *Signal: The Official Publication of the AFCEA*, p21.

12 Peterson, T. "Data Scrubbing." (February 10, 2003) *Computerworld*, 36:6, p32.

13 Fazlollah, Mark and Nicholas, Peter. "U.S. Pads Its Arrest Record on Terrorism." (December 15, 2001). *Philadelphia Inquirer*

14 U.S. General Accounting Office. "Justice Department: Better Management Oversight and Internal Controls Needed to Ensure Accuracy of Terrorism-Related Statistics," January 2003, GAO-03-266. <http://www.gao.gov/new.items/d03266.pdf>

15 Drill Down Demonstration. SAS Institute. <http://support.sas.com/rnd/web/intrnet/demos/dispatch/drill.html>

APPENDIX A: TRAC'S DRILL DOWN DATA MINING TOOL

TRAC's Going Deeper data mining tool can be used for drilling down from aggregate counts down to sets of similar records or further down to details of individual record. Originally developed using SAS Application Dispatcher® code, this tool is also useful for assessing data quality while we process data. The tool consists of four parts: the Going Deeper Main Page, the Drill Down Page, the Listing Page, and the Case Detail Page. Each of these pages is generated by SAS code using either ODS HTML or old-style PUT statements to generate web output.

GOING DEEPER MAIN PAGE

The Going Deeper Main Page (see Figure A-1) is where you select the drill variables, stages to focus on, and statistics to display. As you can see, you can select the drill order as well. When you click the submit button, the receiving SAS program checks variety of conditions, such as whether the data set is a summary data set, whether the data set exists or not, whether drill variables have been selected, whether listing variables have been selected, whether a summary data set has been specified, etc.

The SAS program will generate a report depending on the drill variables. This is done dynamically from the summary data set. The segment of code below illustrates the method used for dynamically generating the where clause. Note that "level" is the current drill down level, &&nvar&i is the macro variable for the name of the drill variable i, and &&val&i is the macro variable value for the name of the drill variable i.

```
/* The dynamic creation of the where clause: */
%do i=1 %to &level-1 ;
    %if &i = 1 %then %do ;
      where &&nvar&i =
            %if &&nvtype&i = C %then "&&nval&i" ; %else &&nval&i ;
    %end ;
    %else %do ;
      and &&nvar&i =
            %if &&nvtype&i = C %then "&&nval&i" ; %else &&nval&i ;
    %end ;
  %end ;
```

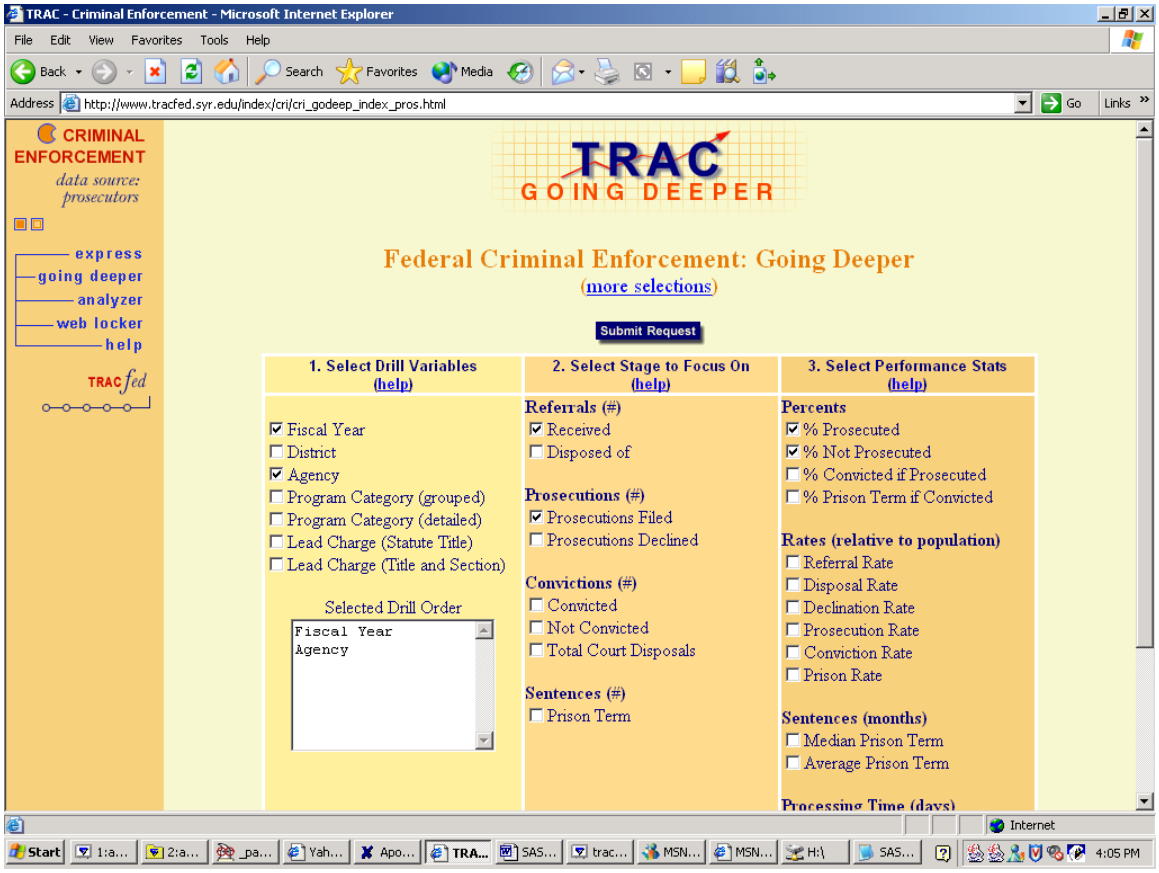


Figure A-1. Going Deeper Main Page

DRILL DOWN LEVEL 1

The screen shot in Figure A-2 shows the first level of the Drill Down Page (level 1). When you click on any year, the next drill down level will be 2 (i.e., level+1). This is done in the back-end SAS program, which creates next level 'link'. A necessary change is the increase in the level number. The following code segment is used to generate the URL that links to drill down level 2:

```
/* cgiscript forms the 'url' link to the next level */
%let cgiscrpt= &_url?_program=&_program;
%let cgiscrpt= &cgiscrpt%nrstr(&)_service=&_service;
%let cgiscrpt= &cgiscrpt%nrstr(&)_debug=&_debug;
%let cgiscrpt= &cgiscrpt%nrstr(&)gdid=&gdid ;
%let cgiscrpt= &cgiscrpt%nrstr(&)_index=5!_logID!classlst!vars!bigstat ;
%let cgiscrpt= &cgiscrpt%nrstr(&)godeeper=&godeeper ;
%let cgiscrpt= &cgiscrpt%nrstr(&)level=%eval(&level+1) ;
%let cgiscrpt= &cgiscrpt%nrstr(&)sumlevel=&type
```

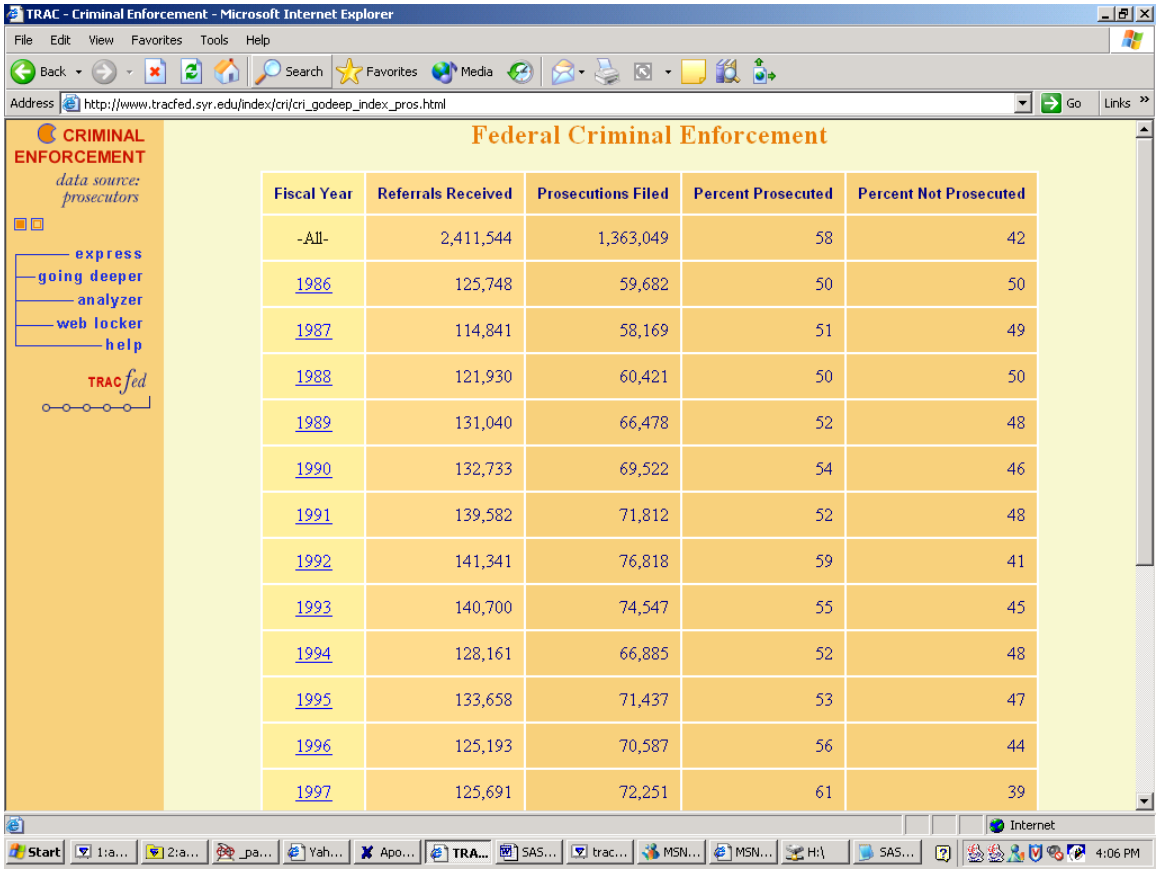


Figure A-2. Drill Down Page, Level 1

DRILL DOWN LEVEL 2

The screen shot in Figure A-3 shows the next level (i.e., level 2) in the drill down. The number of levels is determined by the number of drill variables selected on the main page; in this case, we have reached the end of the drill down at level 2. The links generated will therefore be to the listing program. The following code segment is used to generate the URL that links to the listing program:

```
%let altcgi= &_url?_program=fedcri.sasmacr.detllist.macro ;
%let altcgi= &altcgi%nrstr(&)_service=&_service ;
%let altcgi= &altcgi%nrstr(&)_debug=&_debug ;
%let altcgi= &altcgi%nrstr(&)gdid=&gdid ;
%let altcgi= &altcgi%nrstr(&)_index=5!_logID!classlst!vars!bigstat ;
%let altcgi= &altcgi%nrstr(&)datapath=&datapath ;
```

The following code ouputs the link only if the current level (&level) reaches the number of drill down variables (nclsvars), that is, if we have reached the end of the drill down.

```
if &level = &nclsvrs then do ;
  put "<TD ALIGN=center BGCOLOR=#&classhue><A HREF='" "&altcgi" "&altcgi1" "&altcgi2"
"&bgvals" "&hues" "&detai$
end ;
```

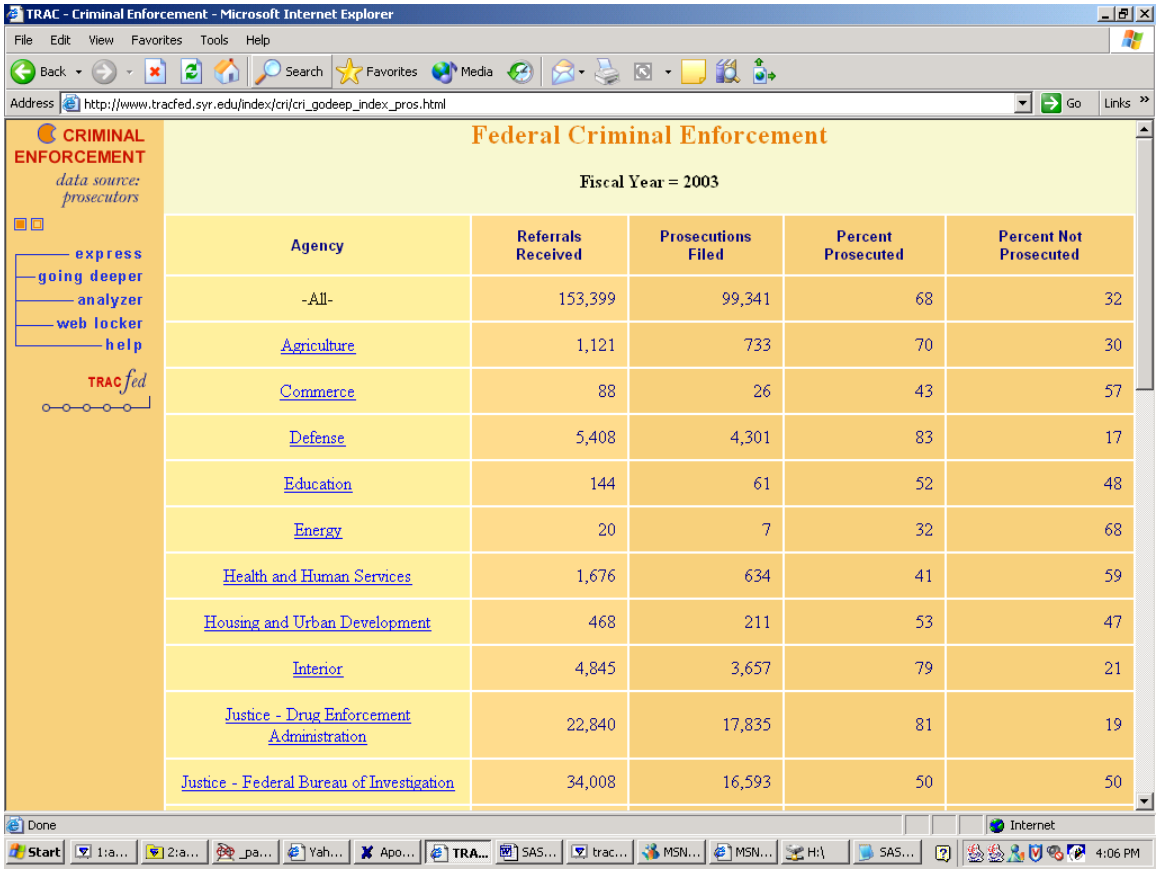
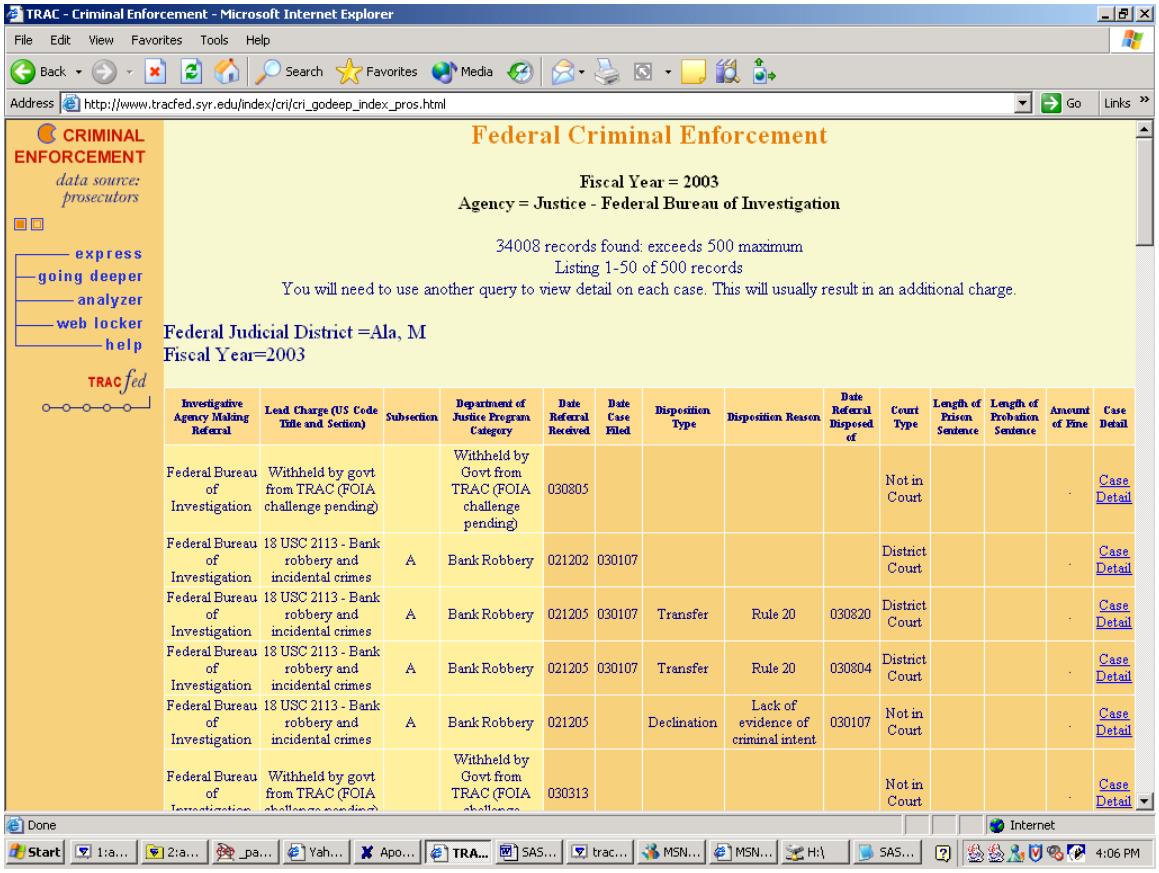


Figure A-3. Drill Down Page, Level 2

LISTING PAGE

A sample of the listing page output appears in Figure A-4. At the end of each listing row there is a link to a more detailed page ("case detail"). Each detail page is identified by the observation number in a link that can be used to execute the case detail display code.

```
detlink="<a href=javascript:winOpen('/cgi- bin/broker?_program=xtools.showrecord.sas"
||"&"||"_SERVICE=&_SERVICE"||"&"||"_DEBUG=
&_DEBUG"||"&"||"godeeper=&godeeper"||"&"||
"gdid=&gdid"||"&"||"obs="||tempobs||"','"||tempobs||tempgdid||"')>Case Detail</a>";
```



Investigative Agency Making Referral	Lead Charge (US Code Title and Section)	Subsection	Department of Justice Program Category	Date Referral Received	Date Case Filed	Disposition Type	Disposition Reason	Date Referral Disposed of	Court Type	Length of Prison Sentence	Length of Probation Sentence	Amount of Fine	Case Detail
Federal Bureau of Investigation	Withheld by govt from TRAC (FOIA challenge pending)		Withheld by Govt from TRAC (FOIA challenge pending)	030805					Not in Court			.	Case Detail
Federal Bureau of Investigation	18 USC 2113 - Bank robbery and incidental crimes	A	Bank Robbery	021202	030107				District Court			.	Case Detail
Federal Bureau of Investigation	18 USC 2113 - Bank robbery and incidental crimes	A	Bank Robbery	021205	030107	Transfer	Rule 20	030820	District Court			.	Case Detail
Federal Bureau of Investigation	18 USC 2113 - Bank robbery and incidental crimes	A	Bank Robbery	021205	030107	Transfer	Rule 20	030804	District Court			.	Case Detail
Federal Bureau of Investigation	18 USC 2113 - Bank robbery and incidental crimes	A	Bank Robbery	021205		Declination	Lack of evidence of criminal intent	030107	Not in Court			.	Case Detail
Federal Bureau of Investigation	Withheld by govt from TRAC (FOIA		Withheld by Govt from TRAC (FOIA	030313					Not in Court			.	Case Detail

Figure A-4. Listing Page

While listing, if the number of rows is greater than 50, then the listing program creates links to each page as shown in Figure A-5 below. This is done by the following code, which calculates and puts the first observation number in the link.

```
/* Page Calculations */
%let firstobs= 1 ;
%let pages= 1 ;
%do %while (&firstobs le &maxobs) ;
   %let page&pages= &firstobs ;
   %let firstobs= %eval(&firstobs+&listobs) ;
   %let pages= %eval(&pages+1) ;
%end ;
%let firstobs= &page1 ;
%let pages= %eval(&pages-1) ;


/* The link is placed for each page */
%do i=1 %to &pages ;
    put "<a href=""&results1" ;
    put "&results2" ;
    put "%nrstr(&)firstobs=&&page&i" ;
```

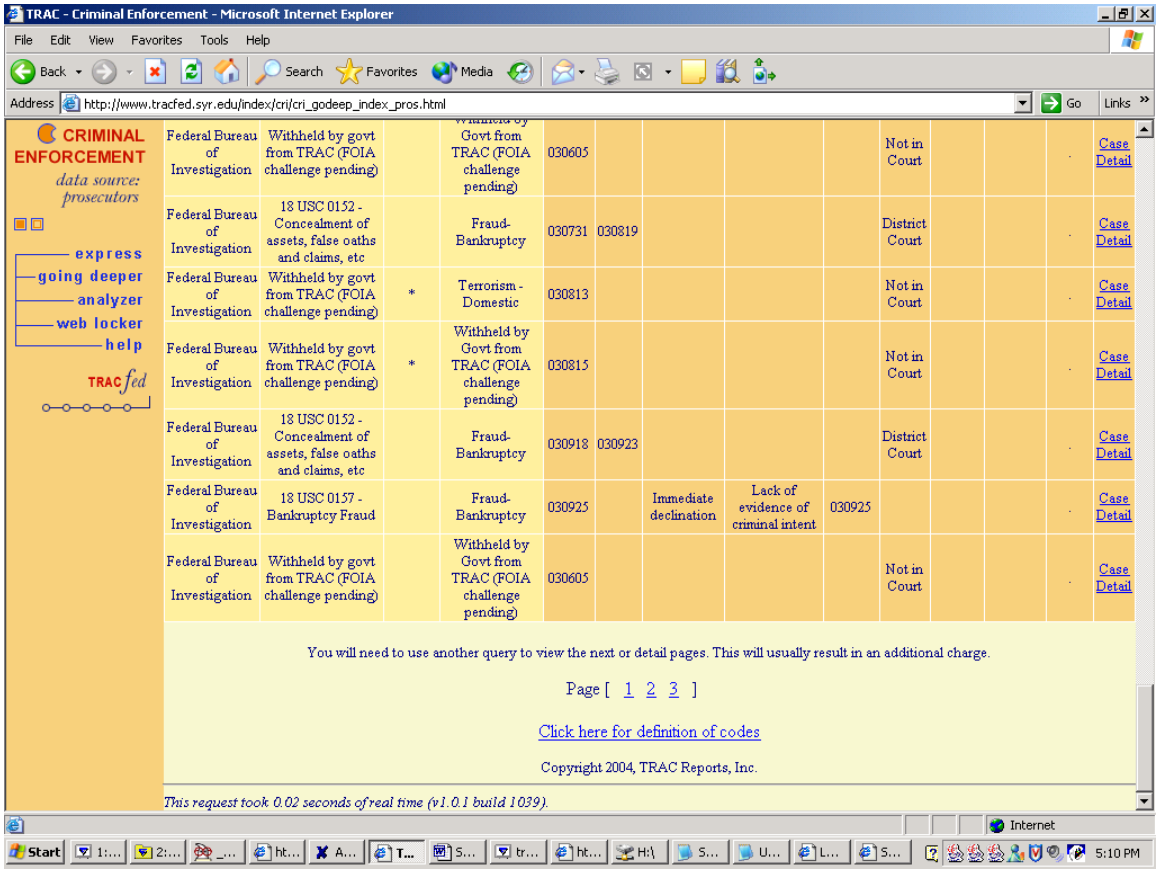


Figure A-5. Page Numbers in Listing Page

CASE DETAIL PAGE

When you click on one of the 'case detail' links, you activate the Case Detail pop-up page, which displays more information about that record. The program that displays the detail page receives that observation number from the 'link' to construct the where clause as

```
%let where=(where=(obs=&obs));
```

This record is then transposed to produce the detail page.

```
/*transpose dataset*/
options missing='';
proc transpose data=record out=record_t;
   var &tvarlist;
run; quit;
options missing=.;
```

The data set 'record_t' is used by ODS HTML and PROC REPORT in producing the HTML page.

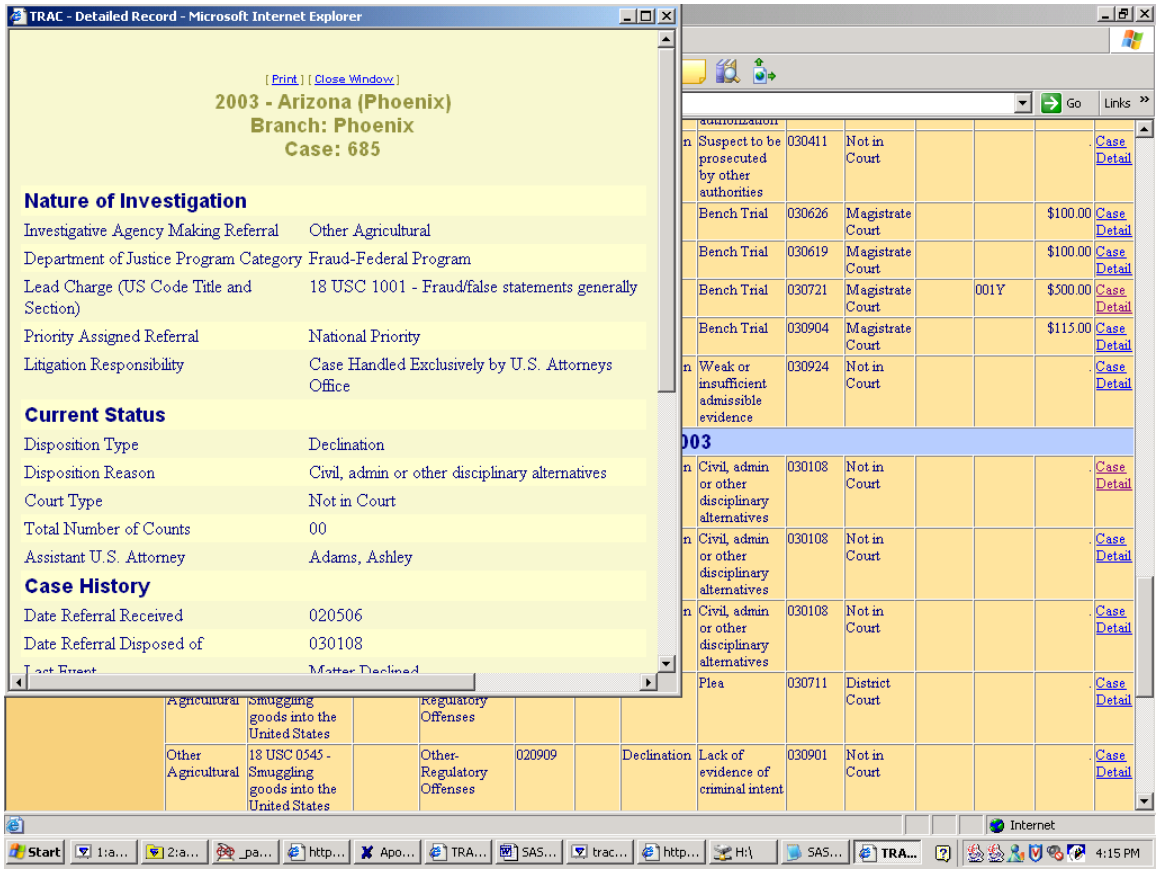


Figure A-6. Case Detail Popup Page

CASE DETAIL ADDITIONAL YEAR

As you can see from Figure A-7, the detail page also provides links to the same record in other years. In this way you can obtain a snapshot of a single record in different years. The following code segment is responsible for creating links to other years. For each year, we create a record in the 'add' data set. This data set is then used for displaying the additional year information. The variables distcode, usaonum and defnum form the key.

```
%let start=1992;
%let end=2003;
%let addwhere=%str(distcode="&distcode" and usaonum="&usaonum" and defnum="&defnum");
%let detlink=%str(detlink="<a href=javascript:winOpen('/cgi-
bin/broker?_program=xtools.showrecord.sas"||" &"||"
_SERVICE=&_SERVICE"||"&"||"_DEBUG=&_DEBUG"||"&"||"lib=&lib"||"&"||"key="||key||"','new"||t
empobs||"')> Case Detail</a>";);
```

The following code creates the data set 'add' with the link to the additional years. If the year is the currently displayed one, then it appears without a link as 'Currently Displayed'.

```
data add;
   length detlink $195;
   set
      %do i=&start %to &end;
         %let addfy=%substr(&i,3,2);
         add.&detail&addfy(where=(&addwhere))
      %end;
   end=last;
   tempobs=put(_n_,z4.);
   &key;
```

```
   if &sort="&fy" then detlink='Currently Displayed';
   else &detlink;
   if last then call symput('addobs',_n_);
run;
```

The following code writes out the links by making use of the 'add' data set created by the previous step.

```
proc report data=add;
   column nothing &addcol;
   define nothing / noprint computed;
   &adddef
   compute nothing;
   count+1;
   if mod(count,2) then do;
      call define(_row_, "style", "style=[background=#FFFFCC]");
   end;
   endcomp;
   title1 "Snapshot of Case Status by Year";
   title2 "<font size="-2">[ <a
href=javascript:winOpen('/help/data/crimcivSnapshots.html','helpWin')>Help</a> ]</font>";
   footnote1 &addfn1;
   footnote2 &addfn2;
run; quit;
```

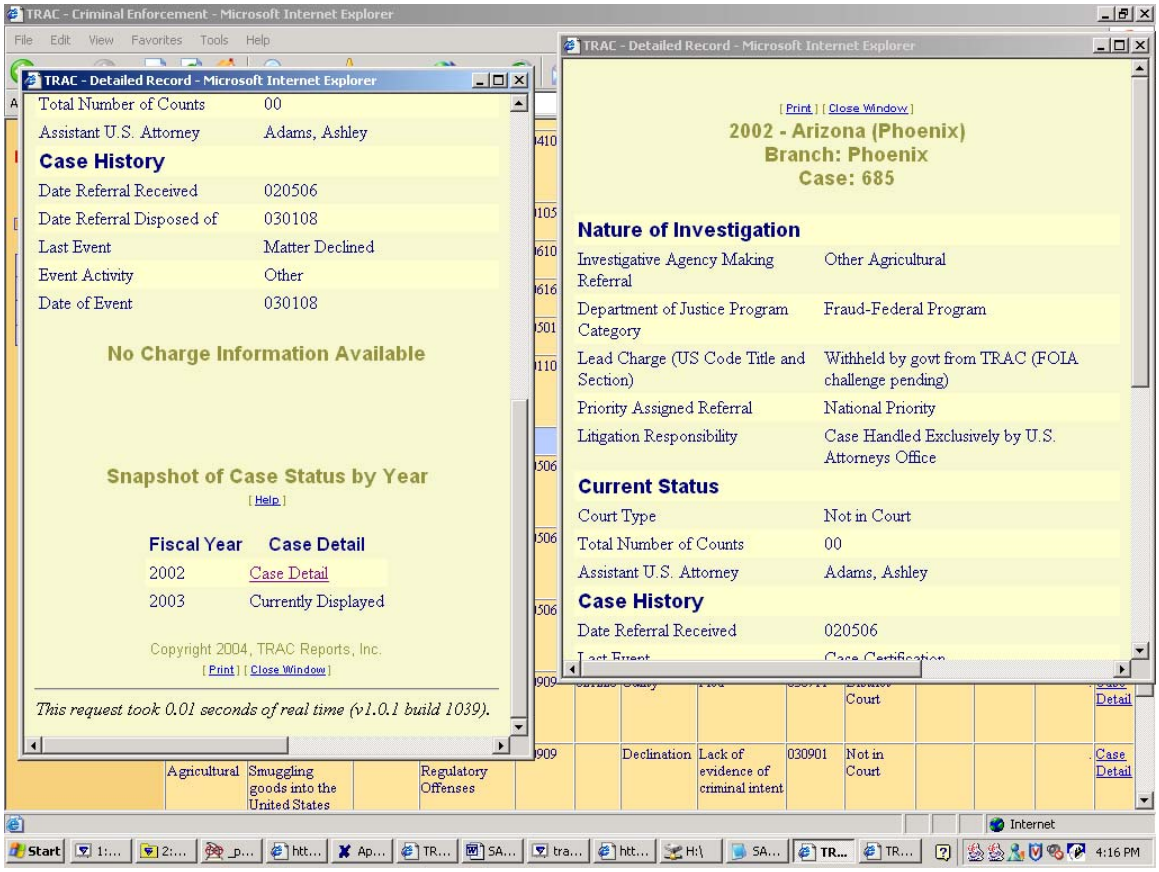


Figure A-7. Case Detail Page for Multiple Years