CHAPTER 5

Data Access and Testing

As pointed out often in this book, data access, verification, and testing are the crucial activities of modern auditors who want to add value to their clients by assessing and guaranteeing the credibility of the information generated or provided by means of computers. To accomplish this, auditors must first apply their critical mentality to the data underlying the information.

For many years, accessing and using data has been the domain of the computer audit specialist, primarily because of the technical knowledge required to use CAATT software. However, developments in computer technology and CAATT software have placed the responsibility clearly on the shoulders of the general auditor. This means that all auditors must have a better understanding of data access methods, data integrity, and the use of CAATTs.

The first part of this chapter discusses the various conditions for accessing data. Then the assessment of the data reliability precedes the decisions about the amount, direction, and intensity of any testing to be performed. This is followed by a topology of data tests that modern analysis software and technology support, and by a discussion of the potential problems associated with the incorrect use of CAATTs.

Data Access Conditions

The client may be internal to the organization, at a regional office, or external to the company. The data may be on a mainframe, minicomputer, or microcomputer system. Regardless, at some point in time, all auditors using CAATTs will be required to obtain access to client data. A problem in accessing client data was one of the main barriers to the widespread use of CAATTs. However, new techniques allow for the efficient and easy access to and transfer of client data. While there is no single method for accessing client data, there exist several possibilities. With careful planning

and a good understanding of the options, most auditors should have little problem in obtaining the data they require.

There are three main options for accessing client data, each with their own problems and opportunities. The first is a traditional one and requires using the client's computer facilities. The second is to download the data from the client's system to the auditor's computer and perform the analyses there. The third is to use mass storage devices (nine-track tapes, cartridges, CD-ROM, optical disks, etc.) attached to a microcomputer that runs powerful audit software.

Prior to choosing between the client's and auditor's facilities, auditors will probably consider the pros and cons of using mainframe versus minicomputers versus microcomputers. Another consideration is the types of electronic data to be accessed. The last and least major consideration is the availability of software for audit purposes.

Mainframe versus Minicomputer versus Microcomputer

One of the first decisions that an internal audit organization considers when deciding to implement CAATTs is: What type of computer is available and should be used? From the early 1960s through the 1980s, there was little choice. The data and the required tools existed only on the mainframe and minicomputers. Auditors would ask the respective programmers to run specialized reports to extract information. In some cases, standard reports were run regularly, while in other cases ad hoc reports were written to satisfy specific audit requirements. In the late 1980s, as a result of increases in the processing speed, storage capabilities, and the development of new tools, the microcomputer became an increasingly viable alternative to the mainframe computer.

In comparing the mainframe, minicomputer, and microcomputer environments, a number of issues should be considered. The following briefly outlines some of the advantages and disadvantages of using the microcomputer for audit purposes. They reflect the mainframe and minicomputer environments and concentrate on recent, major changes in audit technology.

Portability of Programs and Data

Data and audit programs are portable, so the CAATTs and specific analyses can be run or rerun on any microcomputer. The portability of the data and programs is particularly attractive to auditors who are on the road and need to bring the data to the client site or back to headquarters. It is also useful for auditors who are conducting audits at various locations. Standard audit programs (scripts and macros) can be stored onto a microcomputer

and brought to the new site. Data are extracted from the local systems and processed by audit software, ensuring consistency across operations, regardless of the computing environment of the local office. Audit programs are reusable, since no matter where the data came from, the same software is used by the audit team, and the audit programs can also be easily modified.

Limitations to Using the Microcomputer

As the power of the microcomputer increases, the distinction between it and the mainframe becomes less and less clear. Twenty years ago, it was easy to provide a definition that separated the two computer platforms. The old definitions referred to physical size, supported peripherals, and speed. However, the microcomputers of today have become extremely powerful, supporting a variety of peripherals such as tape readers, CD-ROMs, and tape cartridges. Ironically, mainframes have become so small in size that the distinction is further blurred.

More recent definitions attempt to define computers by the function they perform; for example, a microcomputer can be described as:

> *a computer that is part of an individual's office equipment and is used to increase the individual's personal productivity by automating all or part of their work function.*

This definition applies equally to office workers, managers, and internal auditors. In particular, auditors must carry out reviews of the operations of the business. All such operational reviews will contain the same basic steps: planning, research (education), identification of procedures, identification of internal controls (or lack thereof), testing controls, and reporting findings and recommendations. These are the areas where auditors and audit management can use technology to increase individual productivity and organizational effectiveness.

Auditing cannot be automated—it remains fundamentally based on critical thinking. However, many audit functions can be performed more efficiently and effectively with the aid of the computer. Still, many auditors feel that there are a number of barriers to using microcomputers in audit. To address this issue, common concerns are discussed as follows. (See also Will and Brodie [1991], who argue convincingly for the use of microcomputers in auditing.)

Processing Speeds

Similar to the limitations on storage, microcomputers do not process data with nominally the same speed as the mainframe. Jobs may still take longer

to run; however, the rapid development in chip design is constantly improving on microcomputer processing speeds. Using microcomputer audit software, you can already process in excess of 50,000 records per second, and the speed is increasing. The total elapsed time may be less on the microcomputer, since numerous mainframe jobs may have to compete for CPU resources and peripherals.

Single Tasking

In the early 1990s, most microcomputers were single tasking and only one job could run at any one time. The microcomputer was unavailable for other use until the current program had ended. This was a limitation of the basic operating system of the microcomputer. Some earlier operating systems, like UNIX or its derivatives, allowed for multitasking, and today Windows NT and Windows XP operating systems permit true multitasking. For example, a simple script or macro can be written to combine all the monthly files into a single file for the year. The job can run in the background, while the auditor is working on the final draft of the audit report in Word.

Inability to Deal with Complex Data and File Structures

Early audit software was only able to read a few types of files. Today, audit software will accept various types of data (EBCDIC, ASCII, numeric, zoned, binary, Packed, Floating Point, and more). It can read complex file types including variable-length records and multiple record type files, as well as standard fixed-length records. Electronic data is, of course, one of the prime audit objects. The variety of file and data types that can be found in client databases and information systems is immense and can only be understood in an historical perspective. Many organizations have systems that were developed 15 to 20 years ago, and such systems present auditors with unique challenges. The issues are conveniently captured in terms of legacy versus modern data.

LEGACY DATA The term *legacy* data refers to the multitude of files and data types created over the last 30 years by programmers using programming languages such as APL, BASIC, COBOL, and FORTRAN. The data is sometimes contained in diverse Database Management Systems (DBMS)—for example, hierarchical, networked, and relational—and runs on a variety of machines and platforms. Support for such legacy systems may be weak, with corporate knowledge and documentation severely lacking, or even nonexistent. However, legacy systems often perform functions vital to the organization, such as customer billing or payroll, and must be maintained for operational reasons. As such, they contain relevant and useful data for audit

purposes. Since these systems have not been redesigned and rewritten in more modern programming languages, data access and analysis can pose some real problems for auditors. The systems are often not supported by query capabilities and only produce very specific reports or output. Thankfully, modern audit software is capable of handling a variety of data types. For example, data from a system developed on a Unisys system in MAPPER or that uses COBOL repeating fields can be downloaded and analyzed by microcomputer-based audit software.

MODERN DATA The term *modern* data is used to denote the fact that the application system uses a primary data structure (e.g., linear record, relational table, etc.) supported by a DBMS to maintain and administer the data. The systems are also supported by query languages and report writers, making it easier for auditors to access and use the data. However, data variety and inaccessibility are still a problem for many auditors. Interestingly, although identified early as one of the major problems facing auditors (Will and Supper [1975]), many systems are still being developed without any consideration to audit's access requirements. Fortunately, current audit software addresses the problem by providing audit with access to practically all legacy and most modern data structures. Since the data access problems have been largely overcome, auditors can use audit software to perform analyses, whether the data reside on a legacy or a modern system.

Client Facilities

The use of client facilities is an option that should be carefully considered from two viewpoints. The first is one of availability of client software. The auditor may be allowed to use software that already resides on the client's system, which offers some advantages. In particular, the client applications may already have query capabilities that the auditor can use. In other cases, the client may have report writers or specialized software that the auditor can use to access the data directly. For example, the client's application may already be supported by Structured Query Language (SQL) capabilities or an ad hoc reporting function that would make the task of accessing complicated databases easier and less time-consuming. In these cases, the auditor does not have to contend with downloading the data. However, the main drawback is the potential for a loss of independence. If the auditor is using extraction routines or standard reports that were developed by the client, how can he or she be sure of the integrity of the results?

Further, using client software to perform extractions and analyses means that the auditor must have, or develop, sufficient expertise with each client's software. This may require proficiency with a variety of software packages, most of which were not designed specifically for audit purposes and which

may be difficult to use. Should auditors spend their valuable time programming and waiting for debug and test results prior to finally executing the special-purpose program?

A second option is to load the audit software onto the client's computer system. While this addresses some of the earlier concerns such as familiarity with the client's software, it raises new issues as well. The clients may not be inclined to allow unknown software to be loaded onto their computer system. Furthermore, the audit software may not be compatible with the client's environment or may disrupt production jobs. And the auditor may require help in loading the software on the client's system. For these reasons, unless the client site is audited on a regular basis and the software retained on the system, the idea of loading audit software on the client's computer is often not viable.

The use of the client's facilities poses more than a simple problem, and the solution may compromise an auditor's independence to the extent that the data testing may not be undertaken. Fortunately, the option of using the auditor's facilities has improved over the last 15 years.

Auditor's Microcomputer-Based Facilities

The personal computer (PC) has revolutionized, democratized, and globalized modern computing and is no longer the toy it was when first introduced. The PC has become a vital part of many business operations. In fact, coupled with the development of audit software, auditors are now (and for some, for the first time ever) capable of performing their work comprehensively, independently, and professionally.

Rather than performing the analysis on the client's system, often the preferred option is to extract the data to a file and download the file to the auditor's microcomputer. Many computer facilities come with standard utilities that the auditor can use to make copies of data files. However, sometimes it may be necessary to have the client perform the extraction and download. In these cases, the auditors must be confident that the extraction and download has been performed to their specifications. Auditors should critically review the jobs written by the client and, where possible, compare the results of these jobs with control totals, standard reports, or other independent sources of information.

The option of downloading client data to the auditor's microcomputer is becoming much more feasible than it was in the 1980s. Today, microcomputers are better equipped to handle the volumes of data that may be required for a large audit. There exist more options for accomplishing the download. In the early 1980s, the primary option was the use of terminal emulation software such as Kermit or Xtalk and a controller card such as an IRMA board. Now, microcomputers support a variety of transportable

media including DAT drives, 32-channel tape cartridges, memory sticks, external hard drives, and CD-ROMs. Each of these media hold hundreds of megabytes of data, making access to, or the transfer of, even large data files a distinct possibility.

Local area networks (LANs) often have built-in gateways and mainframe terminal emulation capabilities that allow microcomputers to connect directly to the mainframe systems and allow the direct downloading of data files to microcomputers. Once the data is downloaded, microcomputers, especially those with the latest chips, have more processing power and speed than mainframes did 10 to 15 years ago. So there is no need to worry about processing capabilities anymore, considering the variety and power of the software available on microcomputers today.

In the case of DBMS, a number of options can be employed to access directly or create indirectly the relevant data files for downloading. Most DBMS applications have utilities that support the extraction of data to a flat file. Most audit software also supports Open Database Connectivity (ODBC) compliant databases, so SQL queries can be written to extract the data. Some audit software has built-in interfaces to ERP systems and, if all else fails, another option is the generation of a report that is captured in a file and then downloaded and analyzed as if it were a data file. The extraction of the required data no longer remains an issue. However, ensuring the accuracy of the extracted data, once downloaded, is still extremely important.

Data Extraction and Analysis Issues

The first step in employing CAATTs as an audit tool is obtaining access to and analyzing the client data. This includes not only physical and logical access to the data, but an understanding of the data, an ability to use the audit software, and, from time to time, support from technical specialists. All these issues must also be considered in the context of the objectives of the audit.

As recently as 15 years ago, the options for transferring data from mainframe to microcomputer were few and often slow. Communication software brought data down using SNA gateways at 2400 baud. Today, not only have the options increased, but so has the speed at which data can be downloaded. In addition to downloading data, you now have the option of processing CD-ROMs, external hard drives, memory sticks, and tape cartridges directly with the microcomputer. For example, in one organization, a production job copies the inventory database to a file at the end of each month. The file is sent to the internal audit server via file transfer protocol (FTP). Since the inventory system is live, the monthly file provides audit with a snapshot in time and can be used for trend analysis and other tests. In another organization, CD-ROMs, with detailed transactions from the

financial system, are produced quarterly and sent to the audit department. In another, ODBC is used to extract hundreds of gigabytes, which are then stored on an external hard drive.

Accessing the Data

Many audit departments do not have any access to the company's electronic data. With advancements in technology, both hardware and software, this is rarely a technical issue anymore. Audit software can read and analyze most data structures, and microcomputers can handle large volumes of data. Sometimes, lack of access is a result of the client's reluctance to provide audit with access to the application systems rather than a lack of technology.

Support from management may be needed for audit to obtain physical and logical access to the required information. This may require a strongly worded statement from senior management to the effect that "auditors will be given access to any and all application systems and information required to perform their duties." For key application systems, whose information is regularly required by audit, this would mean read-only access at all times. For other less-used systems, the access may be granted on an as-required basis only. For example, audit may always have access to the inventory system, but may only require temporary access to the hiring data in the personnel system while auditing the company's progress toward employment equity in hiring practices.

Audit organizations that have secured management support and direction regarding access to systems and information should ensure that the client is well informed of audit's access rights and requirements.

No matter how the data file is created, the auditor should consider the following when obtaining access to, or downloading, client data:

- Obtain client agreement that the requested data can be used to address the audit objectives.
- Obtain a listing of required tables, key fields, and the logical and physical database structures.
- Obtain copies of record layout and definitions of all fields and ensure that you have a good understanding of the data. The record layout will describe each field and provide information concerning the starting and ending positions and the data type (numeric, packed, character, etc.).
- Obtain a printout of the first 100 records in the data file and compare this to a printout of the downloaded data. This can be useful when building a format file for the downloaded data.
- Obtain look-up tables for all fields stored as coded values and explanations of all possible values for coded fields.

- Obtain the range of valid values and the edit checks for each field. This information can be easily used to verify the contents of each field and test edit checks.
- Verify data for completeness and accuracy, including checking the field types and formats, such as identifying all records with an invalid date in a date field.
- Produce control totals on the mainframe and compare with totals of the downloaded data to ensure all records have been properly extracted and downloaded and that the downloaded file was properly analyzed and interpreted. This establishes the basis for extensive tests of the data.
- Check original reports against independently produced audit information in order to evaluate the extent to which the original purposes have been or are being met.

The main goal is to ensure that the downloaded data file contains all the information needed to perform the audit, that its structure is known, that the files are clean, and that the audit team knows all formalities about them. If these conditions are met, the auditor can safely proceed with an assessment of the integrity of the data.

Data Storage Requirements

It is still true that the hard disks on microcomputers are limited in capacity, although not as limited as ten years ago. Disks of 300 gigabyte (three hundred billion bytes) in size are readily available, even on laptop computers. However, the physical size of hard disks is still fairly limited when compared with mainframe disks. Therefore, considering the vast quantities of data that the auditor needs to analyze, it may not be practical to transfer all of the required data from the mainframe.

Case Study 34: Processing Multireel Volumes of Data

About ten years ago, auditors of a European bank were forced to process millions of transactions stored on many reels of nine-track magnetic tapes with microcomputers and an attached nine-track tape reader, running under innovative audit software. They were asked to prove to top management and to the MIS department that internal audit was and could continue to be technologically independent while saving costs, improving their services, and providing information not available through the MIS department. They did it, of course, using the powerful features of modern audit software. Today, they could do it even more

easily and quickly, because the software has become more user-friendly and handles multivolume and continuous multirecord processing with control breaks and intelligent record selection automatically.

Alternatively, a sample number of transactions can be downloaded and reviewed with the audit software. If the review proves fruitful, a program could be written on the mainframe to perform the same analysis. Some audit software runs on microcomputer, mainframe, or client-server platforms, so the audit program can be uploaded to the mainframe and run, with little or no changes to the code.

Case Study 35: Processing against a Sample File

The inventory system contained millions of transactions, requiring more hard disk space than the auditor had available. CPU time was also very expensive, because a service bureau was used for processing. Rather than using the mainframe to analyze the millions of records in the database, the auditors first extracted a random sample of 200,000 transactions. They used their own microcomputer to test a number of their hypotheses on the sample file. When they had debugged the programs and were satisfied with the results, they uploaded the analyses to run them on the mainframe against the complete database. In this way, they reduced the overall service bureau costs by developing their analyses on the microcomputer and still were able to perform a 100 percent test of the data.

Analysis of Data

Audit software permits auditors to interact directly with the data with minimum knowledge of specialized programming techniques. Most audit software packages have a user-friendly interface and are menu-driven. Auditors can analyze data files using a declarative language rather than a procedural programming language. Certain functions are automated to the extent that one command can be used to carry out a fairly complex task. For example, auditors can calculate averages, means, and other meaningful values simply by running a statistical analysis using audit software (see Exhibit 5.1).

Audit software must be very powerful, allowing fairly complex applications to be written, even to the extent of simulating the processing portions of a production program to verify that processing of the data has been accurate and complete (see Parallel Simulation in Chapter 2).

EXHIBIT 5.1 Statistics on Payment Amount

	Number	Total	Average
Positive:	4,670	15,906,511.96	3,406.11
Zeros:	9		
Negative:	304	−24,780,513.74	−81,514.85
Totals:	4,983	−8,874,001.78	−1,780.86
Abs Value:		40,687,025.70	
Range:		11,248,084.00	

Highest 5: 5,176,542.00, 1,146,200.00, 725,000.00, 360,000.00, 357,000.00
Lowest 5: −6,071,542.00, −5,176,542.00, −4,974,042.00, −1,146,200.00, −1,146,200.00

Risks of Relying on Data—Reliability Risk

CAATTs can be used to improve the efficiency and effectiveness of most audits. With CAATTs, financial, personnel, inventory, and other data can be used to select a sample, perform 100 percent examination of the data, examine trends, and conduct detailed analyses and much more. But obviously, the utility of these tools and techniques is dependent on the integrity of the data. The Canadian Institute of Chartered Accountants has produced a document that discusses the application of audit software in the context of audit risk (CICA [1994]).

The auditor's concerns over data integrity (or lack thereof) will change in complexity and nature, depending on whether the computer application is being audited or the application's data is merely being used to support an audit. The audit of a computer application will typically contain steps to assess the integrity of the application, including the completeness, timeliness, and accuracy of the data. However, there are many times when an auditor is only using the data from an application to review a client's operations. In these cases, the audit program may not include all the audit steps necessary to fully assess the integrity of the application's data.

In either case, the auditor's concerns over data integrity will be proportional to the reliance placed on the data analyses. Therefore, the auditor must assess the integrity of the data before using the data. However, how and to what degree must the integrity be examined? How much is too much (over-auditing), and when is it not enough (under-auditing)? The answers to these questions can be determined by first assessing the risk of relying on the data analyses (i.e., the reliability risk), and second, determining the amount of data testing that must be completed (GAO [1991]).

EXHIBIT 5.2 Factors Affecting the Risk of Relying on the Data

Reliance on the Data	Knowledge of System	Reliability Risk
Sole support for audit recommendations	None	High
	Limited	Medium
	Extensive	Low
Used in combination with other information	None	Medium
	Limited	Low
	Extensive	Very Low
Used as background only	None	Low
	Limited/extensive	Very low

Of course, auditors should never assume that the computer-based data is reliable. Steps must therefore be taken to provide reasonable assurance that the results of the data analyses will be valid. The evaluation of the integrity begins with an assessment of the reliability risk. This risk is dependent on the auditor's reliance on the data and on the auditor's knowledge of the system:

$$\text{Reliance on the Data} + \text{Knowledge of System} = \text{Reliability Risk}$$

The more reliance the auditor is going to place on the results of the data analyses and the less experience the auditor has with the system, the higher the risk of drawing inappropriate conclusions (GAO [1991]). Conversely, the lower the intended reliance on the data, and the better the auditor's knowledge of the system, the less critical the risk becomes. The relationship between these variables can be shown in Exhibit 5.2.

Let us now look in more detail at the two factors determining reliability risk: the auditor's reliance on the data and knowledge of the system.

Reliance on the Data

The first way to reduce the reliability risk is to reduce the auditor's dependence on the data and on the analyses performed. To do this, the auditor should strive to use other information sources, such as management reports and previous audit results, when planning an audit and evaluating the results of any analysis performed. Where possible, the auditor should seek independent verification of the analysis results by reviewing existing reports, such as standard user reports, control totals, exception reports, and error and problem logs. By supplementing the auditor's analyses with other, independent, sources of information, the auditor can increase the reliability of the opinion formulated through the analysis performed for the audit. This will reduce the risk of drawing inappropriate conclusions.

Knowledge of the System

A second way to reduce the reliability risk is to increase the auditor's understanding of the data and the application. An incomplete or inaccurate understanding of the system, its data inputs, and its information outputs can lead to false reliance and erroneous conclusions.

Case Study 36: Debits and Credits

One company's financial system stored all transactions (debits and credits) in a single transaction file. The auditor was examining the debits for a specific account. But instead of selecting only the debit transactions, all transactions for the account were extracted. Three days into the audit the auditor learned of the mistake.

As illustrated in Case Study 36, the main defense against misunderstanding is knowledge. The auditor can gain knowledge of the audited system by reviewing system documentation and by talking to the system's users and programmers. However, this does not guarantee that the data processed by the system are reliable. If the system has limited or no documentation, or if no one seems to know much about it, then the auditor can develop a better understanding of it by working with the data directly. The auditor can review the data by producing high-level summaries or detailed listings of the data. This could include stratifications of the data on key fields to determine the ranges of their values, the production of summaries in tabular or graphical form, and possibly the use of overview reports on the one hand and detailed scans of the data on the other.

Once the reliability risk is established, the auditor must determine the amount of additional data testing that must be performed in order to confirm the risk assessment. This is accomplished by coupling the degree of reliability risk (High, Medium, or Low) with the assessment of the system controls (Strong, Adequate, or Weak). The combination of these two variables will determine the extensiveness of the specific data testing required:

$$\text{Reliability Risk} + \text{Control Assessment} = \text{Amount of Data Testing}$$

If the reliability risk is high and the controls are assessed as being strong, then the amount of data testing would be less than if the controls were assessed as weak (GAO [1991]). In general, the weaker the application's controls, the more testing required (see Exhibit 5.3).

EXHIBIT 5.3 Determining the Amount of Testing Required

Reliability Risk	Assessment of Controls	Amount of DataTesting Required
High	Weak	High
	Adequate	Moderate
	Strong	Low
Medium	Weak	High to moderate
	Adequate	Moderate to low
	Strong	Low
Low	Weak	Moderate to low
	Adequate	Low
	Strong	Very low

Assessment of the Internal Controls

There are two basic approaches to assessing the strength of the internal controls. The first, a system review, directly assesses and tests the controls. This usually involves a review of the general controls and the application-specific controls. A review of the general controls could include reviewing system documentation and the physical and logical security, as well as organizational controls such as the separation of duties. The test of the application controls could include reviewing the source code, verifying input to source documents and output reports, comparing batch and control totals, and using test data, parallel simulation, or other tests.

The second approach, a limited review, involves examining the major controls, reviewing the data for reasonableness, and validating the edit checks. Often a limited review can be performed to determine whether the data can be used for CAATTs purposes. During this type of review, the auditor should endeavor to identify the sources of data errors.

The application's data can be corrupted on input, during processing, or at the output stage. Input errors related to accuracy, timeliness, and completeness can be found by comparing the data to source documents. The processing errors can be identified through parallel simulation of all or certain processes. Output errors can be found by comparing input documents to output reports and through the comparison of the results of the parallel simulation with the system output results.

Critical data testing continues to be one of the main occupations of auditors, whether the data is contained in manual files or electronic systems. However, when using data, auditors must also be careful to distinguish between the amount, direction, and intensity of the data testing.

New Topology of Data Tests

Assessing the reliability risk of an audit in terms of the testing to be done requires that the auditor address the threefold nature of data: its syntactic, semantic, and pragmatic dimensions (Will [1996]). This recognition will guide the auditor in terms of the direction and intensity of the required tests. It will also facilitate the rational definition of tests, their possible automation, their use for different types of audit, and their impact on audit opinions.

SYNTACTIC TESTS Syntactic data tests recognize data as collections of symbols according to bit coding conventions in general and specific data structuring options. Syntactic data errors may be a result of improper data entry (failure of the edit checks) or data manipulations by the application (processing errors). Syntactic tests analyze the data with respect to their internal consistency and coherence. In performing syntactic tests, the auditor may verify that all values conform to the field type (e.g., a date field should only contain valid dates) and sort order. The auditor may even recompute certain derived values such as total price (Quantity * Unit Price). Another example is the equality of debit and credit entries in a double-entry bookkeeping system.

It is important to note that syntactic errors may suggest semantic and pragmatic errors. On the other hand, the absence of syntactic errors does not mean that the data are semantically and pragmatically acceptable and reliable.

SEMANTIC TESTS Semantic tests compare the data with their source documents; for example, verifying transactions against the original vouchers. Semantic testing uses criteria such as adequacy, completeness, timeliness, and accuracy. Typical tests may include testing for gaps or duplicates, calculating aged accounts receivable, or performing tests to establish that all customer names represent living customers and not duplicates or pseudonyms.

Semantic data errors may pass syntactic tests. For example, missing records may not fail the syntactic test for internal consistency, but may indicate problems in the pragmatic domain.

PRAGMATIC TESTS Pragmatic tests seek to verify that the data is a true representation of reality. Not only are the assets properly identified (a valid control number), correctly classified, and valued (agree with source documents), but they are also real assets rather than expressions of nonexistent capital and wealth. For example, sampling may help identify pragmatically

risky data and support confirmations of accounts receivable or physical inventory counts.

The intensity of the testing can be classified with respect to the reliability risk associated with each type of error. For example, the internal control of an application system may be judged to be very strong, but if the source documents contain errors or are not well controlled (e.g., lost or entered twice), then the data still will not have a high degree of reliability. Therefore, the evaluation of data integrity will require the auditor to consider the three facets of data testing—syntactic (internal consistency), semantic (consistency to source documents), and pragmatic (true reflections of reality)—when assessing the reliability risks, examining the strengths of the controls, and formulating an audit opinion.

Reducing Auditor-Induced Data Corruption

To minimize the risk of auditor-induced data corruption, audit software provides read-only access to the files. The auditor may only copy or extract data into special audit files, yet these activities may still result in errors. In general, the more operations performed on such data by the auditor, the greater the chances are of auditor-introduced errors.

Errors can occur when the data is extracted from the application to create a file for further analysis. The extracted data is often converted from one format to another; for example, from a zoned decimal field to numeric or from EBCDIC to ASCII. And errors may be introduced if the auditor incorrectly defines the record layout to the audit software. Possible errors include missing fields, fields defined in the wrong order, incorrect field types, or shifted decimal points. These types of errors will invalidate the results of any analysis performed by an auditor.

As stated before, there are a number of things you can do to help reduce the likelihood of auditor-introduced errors. First, use the application to create control totals, such as total number of records and total dollars, and compare these with the totals calculated using the extracted file. Obtain copies of the record layouts and printouts of the first 100 to 200 records and compare them with the results obtained by the analysis software. Compare auditor-generated reports with standard reports produced by the application. Download the data in its original, native format, without translation, when downloading data from the mainframe to the microcomputer. Most microcomputer-based data analysis packages support various data types including ODBC and mainframe formats. For example, audit software packages also support the automatic creation of file formats for COBOL files and the direct use of dBASE and other types of files, without having to export, extract, or create new file formats.

Potential Problems with the Use of CAATTs

It has been said that, "to err is human; to really foul things up requires a computer." Every day you read in the newspaper about computer errors costing companies millions of dollars. I would contend that, for the most part, these computer errors are human errors, efficiently and effectively carried out by the ever-obedient computer. Thus, the use of CAATTs is not a panacea to all your problems. In fact, improper use of automated tools and techniques can cause their own problems. The key to avoiding errors in the analysis and interpretation of client data is knowledge. This knowledge comes from training, experience, and technical support from others.

Without an effective Quality Assurance (QA) program and adequate training for all auditors, the audit organization is at risk. The four common types of errors made by auditors performing data extraction and analysis are:

- Incorrect identification of the audit population, including missing or extra transactions
- Improper definition of data requirements
- Invalid analysis because of misinterpretation of the data, improper logic, or improper use of the audit software
- Failure to recognize CAATT opportunities, resulting in tasks being performed manually rather than electronically

The following examples highlight these types of problems. By reviewing the types of errors, perhaps you can avoid making similar mistakes.

Incorrect Identification of Audit Population

In reviewing the analyses performed by audit teams, too often it becomes obvious that many auditors have not spent sufficient time on the identification of the audit population during the planning phase. It follows that if the audit population is not correctly defined, any subsequent analysis of the data would not support the objectives of the audit. For example, an audit may not be assessing all financial accounts, or may only be interested in certain types of financial transactions. Therefore, the audit team must establish the criteria that will describe the audit population to be assessed. These criteria will be used to extract the data to be analyzed.

It is critical to develop a good understanding of the data during the planning phase. Discussions concerning the criteria for the selection of the audit population must be given the necessary time and effort to arrive at the right solution. All possible data sources, identification of key fields and their meaning, and timing issues must be resolved before starting the

conduct phase of the audit. The audit teams that use this time wisely usually derive significant benefits from front-end work. Adequate planning helps the team develop an effective data analysis strategy and ensure that they have properly identified the audit population. Failure to do so is illustrated in Case Study 37.

Case Study 37: Financial Audit

In one financial audit, the expenditures at a regional office were understated by more than $5 million. This was the result of the erroneous assumption that all the financial accounts for the regional office could be identified by the first four characters of the financial code. While it was true that most of the financial accounts rolled up, using the first four characters of the regional office account, this was not always the case. The financial system had a more complex structure, and the erroneous assumption falsified the financial picture for the regional office.

In this audit, more than 20 additional financial accounts were missed by the audit team. These could have been identified by using the financial tables and performing a variety of searches, including financial account code, location, and financial account manager. Alternatively, had the auditors checked with the controller, they would have been given the information they required. The failure to correctly identify the audit population resulted in the conduct phase taking twice as long.

In Case Study 38, a sample of personnel was selected for several regional offices. However, the auditors incorrectly defined the audit population from which the sample was selected.

Case Study 38: Personnel Audit

The Corporate Human Resource Information System (CHRIS) contained at least three fields about the employee's location, including the:

- Administrative location against which the position was charged
- Physical location of the employee (where the employee worked)
- Reporting location (where the personnel files were kept)

The auditors assumed that the physical location field "LOC" would identify all people with personnel files located at the regional office.

Since they incorrectly used the LOC code, they failed to include people working at the office, who had a different LOC code, and included people physically located at the regional office but with personnel files elsewhere. However, the auditors were not aware of these anomalies, so they drew a sample of personnel. The result was that some of the required physical personnel files were located at another site and could not be verified during the on-site file reviews. The missing personnel files caused a lot of problems when the auditors tried to extrapolate the results to the entire population.

Without a proper analysis of the audit population during the planning phase, inaccurate or incomplete data can be used during the conduct phase and produce invalid results. Closely associated with the identification of the audit population is the proper definition of it.

Improper Description of Data Requirements

Almost all CAATTs require access to client data files. Many audit teams have encountered problems when attempting to access the client's data because they failed to properly identify or describe their requirements. The problems include:

- Incorrect statement of audit data requirements, which can result in the information received not being what was needed
- Failure to identify important fields, requiring the auditors to ask for a second or third extraction
- Failure to consider the client's information system (IS) support unit's operational constraints or not realizing that the audit organization has already placed other requests with the IS support unit; these failures make any subsequent dealings with this support unit very difficult
- Requesting information that the audit organization already has or that is readily available from another source
- Obtaining the information in a format that is not conducive to further electronic use
- Failing to tell the client the file format for the requested data file (such as ASCII, flat file, delimited, dBASE, fixed record length, etc.)

Auditors must be careful to define their requirements and to determine the best possible source of data to address those requirements. In many cases, the answer can be obtained from a variety of sources: one or more

information systems, electronic reports, or files already extracted and available within the audit organization, such as summary extractions.

In one organization, the client provided information that stated that the budget for the financial account was $100 million. In fact, the budget was $200 million. In another example, the client stated that the company had spent $33 million on air travel, whereas the auditors determined that the actual amount was closer to $54 million. In both these cases, the differences were attributed to incorrect definition of the auditor's requirements. But a great deal of time was spent reconciling the differences in the numbers.

The auditors will not possess knowledge about every system in an organization. Many audit organizations have addressed this problem by creating an internal Information Support Analysis and Monitoring (ISAM) section, which is responsible for assisting auditors in defining their requirements. Part of the ISAM mandate is to be the focal point for all information requests. This will ensure that the ISAM staff and, indirectly, the entire audit organization, continually improve their knowledge of the company's main application systems. As a result, all audit teams should be encouraged to not only use the support section as a focal point when requesting information, but also to provide feedback to the ISAM staff on data or systems that may be of use to other audits.

Often when the IS audit teams perform detailed analyses of the company's main information systems, little information flows back to the audit organization. All team leaders and audit managers should ensure that audit teams, especially IS auditors reviewing applications, provide feedback to the audit department at the beginning, during, and at the end of the audit. The feedback can be formalized by creating a lessons learned database and/or involving the ISAM staff in the audit.

Invalid Analyses

Since audit software has become easier to use, auditors must be even more careful that the analyses are properly conducted. The auditor cannot assume that the successful completion of a command or operation means that the results are correct. The command or operation must be executed in such a way to produce the desired results. This requires the auditor to have a higher degree of familiarity with the software than simply knowing that they must, for example, click here to join two files.

The computer does what it is told and not necessarily what you meant to tell it. Thus, when auditors are performing detailed analyses, there is always a risk of making an error. Common types of errors include:

- The incorrect definition of the format files for the client data, such as a shifted decimal point.

- The improper inclusion or exclusion of data. For example, audit teams have used duplicate copies of data files in addition to the originals, double counting all transactions. Others have used records in their calculations that had been marked for deletion in dBASE files, but not physically removed because the databases had not been packed. Another frequent error is one of timing, the inclusion or exclusion of data because of cut-off errors and improper syntactic tests.
- The incorrect interpretation of the data. Many audit organizations have seen cases where auditors have made incorrect assumptions concerning the data. The industry is full of examples where an entire series of accounts or insurance policies were omitted from the audit sample because the auditors assumed all accounts started with a number, failing to review a new series of accounts starting with a letter. Inventory turnover calculations have been botched by selecting the wrong date fields, and numerous other errors have occurred in analyses performed by audit teams that neglected to perform proper semantic tests.
- Analyses that are incorrect because the recalculate option was disabled on the spreadsheet package, formulae were incorrectly defined, or the purposes of the information were improperly defined for pragmatic tests.
- The incorrect use of "AND," "OR," "NOT," and other logical relations.
- The incorrect joining of two or more files. This function often has five or more different output results depending on what the user specifies should be done with the unmatched transactions. Also, the results will differ depending on which is the primary and the secondary file and whether it is a many-to-one or a one-to-many match.

A solution to reducing these types of errors is to seek independent verification of results and to compare actual results with expected results. During the planning phase, all proposed analyses should be properly defined in an analysis plan and reviewed for completeness, accuracy, and proper data sources during the conduct phase.

Failure to Recognize CAATT Opportunities

It has been said that, "if the only tool you have is a hammer, all your problems will look like nails." CAATTs give you a powerful tool, so you can use the right tool for the right job.

The failure to recognize opportunities for the use of automated tools and techniques is the biggest single barrier to the successful implementation of CAATTs. Too often auditors have spent days—in some cases, even weeks—manually performing tasks that could be performed by the

computer in a matter of minutes. In one case, the audit team spent several days just trying to list the data in a format that would allow them to visually compare the data with the data in another file. Using the computer to join the files together would have saved the audit teams many hours of work. While the manual comparison of two files could have missed records, the electronic comparison would be 100 percent accurate and could easily be repeated.

Not everyone is expected to have the same level of expertise with audit software tools, but auditors should question any manual manipulation of data that involves matching, sorting, searching, or repetitive calculations. The human brain is ideally suited for tasks that require intuitive logic, pattern recognition, and logical jumps, but the computer is better at routine and repetitive tasks.

Audit organizations that create an ISAM section and use this section to assist auditors in developing analysis plans will find that the number of missed CAATT opportunities will dramatically decrease.

Summary and Conclusions

Many of the early challenges surrounding access to client data have been resolved. It is easier to access client systems, extract client data, and transfer those files to the auditor's microcomputer. The microcomputer, in turn, is more powerful and capable of handling vast amounts of data. What remains is the requirement for the auditor to understand the data, the key fields, and the analysis requirements.

Audit managers should be aware of the potential loss of time due to incorrect or invalid analysis. They should ensure that each team is given adequate time to plan for the use of CAATTs. The planning phase should include the steps necessary to identify possible data sources, to identify the criteria that define the audit population, and to develop an analysis plan. History has proven that audit teams achieve better results when they are supported by someone with audit and computer expertise from the outset of the audit rather than at a later stage.

Team leaders should encourage team members to adequately document the analyses performed and to share ideas and methodologies with others. In particular, any manually performed analysis should be examined closely to determine if it could not be better performed in an automated fashion. Given the myriad of sources of information, team leaders should endeavor to consult with other auditors or ISAM staff to determine the possible sources of data to support the objectives of the audit.

Team members should review the contents of data files and develop a good understanding of the data prior to embarking on long, detailed

analyses. The use and supervisory review of analysis plans should be encouraged. These plans detail all the analyses to be performed and their expected results and will help reduce invalid analyses. The use of CAATTs can bring about significant improvements in the efficiency and effectiveness of many types of audits. However, the extent to which they can be used effectively must be tempered by knowledge of their limitations and the issues affecting the individual audit.

The integrity of the data may be the most significant, single limitation to the use of CAATTs. Only data testing will provide the auditor with a good measure of the integrity of the data. The auditor can then determine whether automated tools and techniques should be used to assess the information generated from them.

As part of the implementation and use of CAATTs, the auditor must ascertain the degree of reliability, the extensiveness of the integrity testing to be performed, and ways to reduce errors. There may be times when a 15 percent error rate in the data does not adversely affect the audit results, but there will be other times when a 3 percent error rate invalidates the audit findings. However, what constitutes an error is not a trivial question but one that requires a three-dimensional approach: syntactic, semantic, and pragmatic.

By following the guidelines in this chapter, the auditor can reduce the likelihood of over-auditing the data where the controls are strong and the integrity is judged to be high, or under-auditing the data where the integrity is questionable. Prudence and probity must be the keywords when using and relying on CAATTs. By creating an environment that recognizes not only the potential of CAATTs, but also the unique challenges and requirements of CAATTs, audit can be successful in the implementation and use of automated tools and techniques for data access and testing.