TR 81-012

COMPUTERIZED ARCHAEOLOGICAL SPECIMEN-CATALOG DATA STORAGE AND RETRIEVAL AT THE UNIVERSITY OF NORTH CAROLINA, CHAPEL HILL

by

Dawn Elizabeth Janney

A thesis submitted to the faculty of The University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in the Department of Computer Science.

Chapel Hill

1981

Approved by:

Stephen F Weiss

Dr. Stephen F. Weiss, Adviser

Dr. Joffre L. Coe, Reader

Dr. Henry Fuchs, Reader

DAWN ELIZABETH JANNEY. Computerized archaeological data storage and retrieval at the University of North Carolina, Chapel Hill. (Under the direction of Stephen F. Weiss.)

SPECS is a computerized system to store and retrieve archaeological data from the University of North Carolina at Chapel Hill's Research Laboratories of Anthropology. It uses interactively entered data to update a tape-base master file of specimen numbers, descriptions, counts, weights, and collection locations. Over 1,000,000 specimens, which can have non-unique identifications, may be entered.

Validation during data entry and file updates verifies data consistency within and between specimens. Data retrieval from variable-length records of variable-length fields uses SAS, a report-generating and analysis package. Passwords and backup files protect data privacy and security.

This document is a detailed description of SPECS' design. Appendices specify the data and record formats, reports, and algorithms that will be needed when SPECS is implemented.

ACKNOWLEDGMENTS

I wish to thank Dr. Trawick Ward, Mr. Jack Wilson, and the members of the Research Laboratories of Anthropology (RLA) at the University of North Carolina, Chapel Hill, who patiently answered my many questions about their laboratory procedures and encouraged me in my work. I am particularly indebted to Dr. Ward for the specimen-catalog information listed in Appendix G, and for his comments on an earlier version of this thesis.

I am also grateful to the members of my committee for their encouragement and advice.

To the extent that SPECS is well designed, it is due to these people; any errors in fact, judgment, or exposition are my contributions.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter I

INTRODUCTION

1.1 DESCRIPTION OF THIS PAPER

This document presents a computerized system for storing and retrieving archaeological data from the specimen catalog of the Research Laboratories of Anthropology (RLA) at the University of North Carolina at Chapel Hill. It is intended to be read by data processors and archaeologists with at least introductory classes in computer programming and systems design.

Chapters 1 through 3 summarize present specimen-catalog data collection, storage, and retrieval techniques, and introduce SPECS, a computerized storage and retrieval system for specimen-catalog data. Chapter 1 is an introduction to the specimen catalog, its users at the RLA, and their problems. Chapter 2, intended to provide background information for readers who are unfamiliar with the RLA, is a description of collection and recording techniques for specimen-catalog data. Chapter 3 characterizes the uses of SPECS and summarizes the priorities set by its users, then concludes with a general description of SPECS.

Chapters 4 through 7 are a more detailed description of SPECS. They present its files, interactive sessions, batch updates, and data retrieval.

Chapters 8 and 9 describe protecting SPECS' files from damage and unauthorized use.

Appendices A through G provide highly detailed information about SPECS. Appendices A, C, and D are for the same audience as the rest of this paper. Appendices B, E, and F will probably be of most interest to systems designers and implementors, and Appendix G, which enumerates the specimen-catalog entries studied for this paper, will be of interest to RLA archaeologists.

1.2 THE USERS

The Research Laboratories of Anthropology (RLA) was established in 1939 as a part of the University of North Carolina at Chapel Hill (UNC-CH) [reference 8]. Its director reports to the Dean of Arts and Sciences at UNC-CH. The RLA offices and laboratory are in the basement of the Alumni Building on the UNC-CH campus; permanent collections are stored in Durham, North Carolina.

The RLA has four full-time employees: the director, Dr. Joffre L. Coe, who is also a professor of anthropology at UNC-CH; Dr. Trawick Ward, the director of computer applications; Mr. Jack Wilson, the laboratory director; and Ms. Estella Stansbury, the RLA secretary. The RLA also has an average of six part-time research assistants during the academic year, and about thirty additional employees during the summer field season.

RLA collections include over 1,000,000 archaeological specimens from the southeastern United States. About 5000 new specimens are added each year.

1.3 THE PROBLEM

The specimen catalog describes each specimen in the RLA collections. The entry for each specimen consists of a unique specimen number and a brief description of the specimen and where it was collected. Information in the specimen catalog is collected during preliminary laboratory processing. Although the RLA may use the specimens later in more specialized analyses, the results of the analyses are not entered in the specimen catalog.

The RLA uses specimen-catalog data to find the specimen numbers of specimens with known characteristics and for analyses of specimen distributions or summaries of specimen characteristics.

Finding numbers of specimens with known characteristics is done by manually searching the specimen catalog for entries for the specimens. This technique is not too inconvenient if a small number of specimens with entries in known locations in the catalog are needed, or if all of the specimens are from a few locations (catalog entries are sorted by specimen number, which is assigned roughly sorted by collection location), but becomes almost prohibitively time-consuming if numbers for many specimens with entries at unknown or widely scattered locations in the catalog are needed.

Using specimen-catalog data for summaries or analyses presents the same problems in finding the appropriate entries as does searching for specimen numbers. Additional problems result from the lack of a standard machine-readable format for the data. An archaeologist who wants to process specimen-catalog data with a computer must either design his own data representation and enter the data, or supplement data already recorded in a format designed for another specific application. The lack of a standardized machine-readable format makes pooling data entered for separate purposes awkward.

These problems suggest that specimen-catalog data would be more accessible if it were stored in an appropriate machine-readable format. Computerized searching could replace the manual searching needed to find desired specimen-catalog entries, and the effort spent in duplicating data encoding and entry could be put to more productive use.

Chapter II

SPECIMEN-CATALOG DATA COLLECTION AND RECORDING

The RLA specimen catalog results from more than forty years of collecting samples from archaeological sites, analyzing them, and recording them as specimens. This chapter summarizes the collection and analysis techniques used to build the specimen catalog, and characterizes their results.

The chapter is neither definitive nor comprehensive. Rather, it is a description, for archaeological laymen, of some of the sources and extent of the complexity of specimen-catalog data. Important techniques and sources of information, such as mapping and feature description forms, that do not directly affect the specimen catalog are not mentioned. The chapter oversimplifies descriptions of collection locations and omits some of the steps in laboratory processing.

The principles guiding RLA field and laboratory methods are implicit in the technical decisions made by RLA archaeologists, but are largely unwritten. New archaeologists being trained in RLA methods serve an informal apprenticeship, in which they gradually are given more responsibility for making decisions as they demonstrate an intuitive understanding of the principles used by the

RLA. More skilled archaeologists modify the existing data collection and recording techniques as needed for new kinds of data, while adhering to precedents set in earlier processing. The consistency of RLA techniques does not always extend to spelling and recording formats, which often vary between individuals. Although this variability is largely unnoticed by archaeologists, it greatly complicates deriving a consistent machine-readable format for specimen-catalog data.

2.1 COLLECTION

Archaeologists collect samples, such as stone tools, pottery, animal bones, nutshells, and glass beads, from archaeological sites. These samples help the archaeologists understand the people who used the sites. Information about where a sample is collected is as important to the archaeologists as the sample itself, for without it they cannot understand how the samples came to be where they were found.

2.2 SURVEYS AND EXCAVATIONS

Most archaeological samples are collected during surveys or excavations. Surveys locate the archaeological sites in an area, describe them, and estimate what kinds of information would be found if they were excavated. Only a few samples are collected from each site when it is surveyed.

Each North Carolina site that is surveyed is assigned a unique number used to identify all collections from the site. A site number is composed of a two-letter county abbreviation, a one-letter code indicating a type of site, and a sequence number within the county, and may also include an additional letter. Appendix G includes many examples of site numbers.

Excavations require digging into the sites to find what lies beneath their surfaces. An excavation of a single site may require several summers' work by a crew of ten or more people. Collections made during excavations are more catholic than those from surveys, and may include over 20,000 samples from one site. Meticulous notes are made during excavations, with particular attention to recording where each sample is collected.

Because the variability in specimen-catalog entries for samples collected during surveys is usually encompassed by that of entries for excavation samples, the rest of this chapter emphasizes excavation techniques and samples.

2.3 LOCATION DESCRIPTIONS

2.3.1 Coordinate systems

The RLA uses a rectangular coordinate system to describe locations within each site. The system, established when the site is first excavated, divides the site into standard ten-foot squares called "grid squares."

The coordinate system has three axes: north-south, east-west, and vertical. The axes intersect at the site's "datum," a specified point on its surface. Horizontal positions in the site are described by their distances in feet from vertical planes along the north-south and east-west axes. Vertical positions are distances in feet below a level plane a fixed distance above the datum.

Horizontal positions are denoted by a three-part notation. The first part, a number, is the distance north (or south, if the distance is negative) of the east-west axis. The second and third parts, a single character and a non-negative number, denote the position east or west of the north-south axis. The character is an "L" for locations west of the north-south axis (to the left of a north-facing observer) and an "R" for locations on or east of the axis (to the observer's right). The non-negative number is a distance from the north-south axis. The site datum, at 0R0, is 0 feet from the east-west axis and 0 feet east of the north-south axis.

2.3.2 Units

Each excavated site is divided into units. A "unit" is a location such that all samples from it are considered to have been collected from one place, different from the place at which any other sample was collected. Samples from each unit are kept separate during excavation and laboratory processing.

2.3.2.1 Natural locations and grid squares

The RLA uses a combination of natural and arbitrary boundaries to divide a site into units. Natural boundaries follow differences in soil characteristics to define "natural locations" within the site. Each natural location has a history distinct from that of the soil surrounding it. Although natural locations may be subdivided by arbitrary boundaries at the archaeologists' discretion, units rarely contain parts of more than one natural location. Because natural locations were not made for the archaeologists' convenience, a system of units based on them is awkward to describe in the site's coordinate system.

Natural locations typically represent the remains of identifiable human activities, such as setting posts in the ground or digging graves or fire pits. They often appear as the interiors of filled-in holes. Common examples of natural locations are "post holes" (in which wooden posts were set, often as parts of buildings or palisades), "burials" (graves), and "features" (the remains of such things as fire pits, that were made when the site was used but are neither post holes nor burials).

The sizes and shapes of natural locations frequently are irregular. A single location may change size and shape at different depths in a site. Although the top of each location was at the surface of the site when the location was formed, later changes in surface level and disturbance of surface soil mean that the location may not extend to, or be recognizable at, the top of the site. If the activity forming one natural location infringed on an earlier location (a fire pit was dug on top of a forgotten burial, for example), the location formed by the more recent activity at least partially obliterates that formed by the earlier one. Thus, one natural location may wholly or partially overlie or truncate another.

The part of the site that is not in any natural location is frequently divided along grid-square boundaries, resulting in locations that are also called "grid squares."

2.3.2.2 Surface, zones, and levels

Each location defined by natural and arbitrary boundaries is divided into one or more roughly horizontal layers. These layers, representing vertical positions, are of three kinds: surface, zones, and levels. "Surface" is the ground surface of the site when excavation is begun. "Zones" have boundaries, defined by changes in the soil, that represent changes in the archaeologically interesting use of the site. "Levels" have either boundaries defined by the edges of soil layers in the "plow zone" (soil disturbed by farming on the site) or arbitrarily located boundaries at depths designated by the archaeologists. Thick zones may be subdivided into levels with arbitrary boundaries; levels cannot be divided into zones.

The relationships between natural locations, the grid squares outside of them, levels, and zones are complex. Features and burials are divided into zones, which may be subdivided into levels. Grid squares may be dug in levels, zones, or levels in the plow zone and zones below it, depending on their soil.

The smallest divisions into levels or zones in the smallest subdivisions of natural locations or grid squares are units.

2.3.2.3 Special cases

Although most units can be defined by a combination of natural locations, grid squares, surface, levels, and zones, a wide variety of special cases cannot. These units vary in shape, size, and relationship to other units. "Point proveniences," "test squares," "test pits," and "multiple-location units" are recognizable categories of special cases. Point proveniences are measurements of precise locations where individual samples were collected from within other units. Test squares are like grid squares, but are not part of the site's coordinate system. Test pits are holes dug into the bottom of the lowest currently excavated level or zone of a grid square or test square. Multiple-location units combine samples from several already defined units.

2.3.2.4 Nomenclature

The nomenclature for distinguishing units is as complex as the ways of defining their boundaries. Features and burials are typically designated by "feature" or "burial" and a unique integer, sequentially assigned among the features or burials in each site, although letters may be added to burial numbers to distinguish individuals buried in a common grave. Post holes are known by "post hole" and a number assigned within the ten-foot grid square in which they occur. Grid squares are known by the coordinates of their southeast corners.

Zones are numbered from the top of the site downwards; since zones are natural soil layers, the designation for each zone is constant and unique across the natural location (or part of the site outside all natural locations) in which the zone occurs. Levels in the plow zone are numbered from 1 to 3A (between 3 and 4); levels below the disturbed soil at the top of the site are numbered in ascending order proceeding downward from level 4. Levels within zones are designated by letters, begining with "A" at the top of each zone.

Point proveniences are designated by their horizontal and vertical locations, expressed in the site's coordinate system. Test squares are distinguished by single characters, uniquely assigned within each site. Test pits are numbered sequentially within the grid or test square in which they are dug. Multiple-location units have designations improvised to fit the units, but are commonly known by ranges of coordinates or a list of the units composing the multiple-location units.

Regardless of the kind of location, the most important rule of nomenclature is that enough information must be given to uniquely identify each unit. Other rules are also generally followed: spelling and abbreviations follow the judgment of the archaeologist describing the unit, redundant information may be given, and location descriptions may be supplemented by detailed descriptions of anything the archaeologist wants to include.

2.4 LABORATORY ANALYSIS

The standard laboratory analysis procedure at the RLA can be characterized as a process for converting samples into specimen catalog entries and specimens ready for permanent storage. The three steps of sorting samples by location, recognizing specimens, and cataloguing are described below.

2.4.1 Sorting by location

Sorting the samples orders them according to the locations of the units from which they came. The sorting order for samples from excavated sites is complex, and currently involves four levels of keys.

The first two key levels are field season and site identification. Sorting by field season is chronological, and simply reflects the order in which the samples were collected. Sorting by site involves grouping all samples from each site, but the order in which sites are processed is often irrelevant when processing is finished.

The third level of keys is the location within the site. This level comprises a large number of kinds of key values, each defined for only part of the samples. Grid square designations, post hole numbers, feature numbers, and burial numbers are examples of keys at this level; most specimens are sorted by at least one of these examples.

Grid square designations are used as keys only for specimens collected in post holes or outside of natural locations. The extent to which grid squares are used in sorting depends on when the sorting was done. The current RLA practice is to sort grid squares into southwest-to-northeast order with the north-south coordinates changing less rapidly than the east-west coordinates (analogous to row-major order for a two-dimensional array imposed on a site map with north upwards); this policy is followed in recent samples like those from the 1976 field season at site SkV1a. Older samples may be sorted into other orders, or not sorted by grid squares at all.

Samples from post holes, features, and burials are sorted by increasing post hole, feature, or burial number. A grid square may be indicated for samples from features and burials, but is redundant and not used as a key.

The fourth level of keys is vertical location: surface, level, or zone. Samples are sorted in order of increasing collection depth (surface, then increasing level or zone number, with designations for levels within each zone sorted into increasing alphabetical order).

2.4.2 Recognizing specimens

Recognizing specimens is done by combining the samples from each unit and separating them according to kind. The resulting groups, each containing all of the material of the same kind from the same location, are called "specimens."

Some specimens are divided into "subspecimens," which often have different uses in more sophisticated analyses. Specimens composed of potsherds are often divided into "large and rim sherd" and "small sherd" subspecimens, for example, and specimens of glass beads may be subdivided by color.

2.4.3 Cataloguing

Cataloguing a specimen is creating a specimen-catalog entry for it. Cataloguing is done in the four steps of sorting by kind, assigning a specimen number, counting and weighing, and recording. These steps may be overlapped or reordered to suit the convenience of the archaeologists.

2.4.3.1 Sorting by kind

All specimens from each location are sorted by the kind of specimen they are, according to the order in which they are listed in Table 1. The RLA includes specimens of kinds not listed in the table by using a series of implicit sorting conventions that perpetuate the grouping of similar specimens. This sorting by kind automatically places each specimen into one of the six types denoted by the one- or two-character codes shown in figure 1.

2.4.3.2 Assigning a specimen number

The RLA assigns each specimen a unique "specimen number" created by concatenating an "accession number," a "specimen type code," and a "sequence number." The accession number always is a positive integer, the sequence number usually is, and the type code is one of the types listed in figure 1. In the specimen number "2317p102", for example, 2317 is the accession number, "p" is the type code, and 102 is the sequence number.

The accession number for a specimen is determined by the site and year in which the specimen was collected. The relationship between site, year, and accession number is complex: there may be many sites with the same accession number, several accession numbers per site, or several years' collections from one site with the same accession number; but there should never be more than one accession number for a single site and year.

TABLE 1

Specimens and types for sorting by kind

Specimen	Type
Projectile point (C.S.P.P)	a
Biface	a
Drill	a
Hoe	a
Used flake	a
Grinding-hammerstone	a
Pestle	a
Axe (celt)	a
Stone disk	a
Pendant	a
Stone pipe	a
Clay pipe	a
Clay disk	a
Figurine	a
Pot	a
Cup	a
Spoon	a
Turtle shell cup	a
Awl	a
Scraper	a
Shell bead	a
Shell scraper	a
Shell pendant	a
Iron hoe	a

Iron axe	a
Copper bead	a
Trade bead	a
Potsherd	p
Steatite sherd	p
Historic sherd	p
Animal bone	b
Charcoal	eb
Seed	eb
Tooth (human)	h
Flake	m
Raw material	m
Ocher	m
Daub	m
Coil	m
Shell	m
Soil sample	m
Washings	m

The specimen type code is "a", "b", "eb", "h", "m", or "p." It is redundant, in the sense that specimen numbers that cannot be uniquely identified by their accession and sequence numbers arise only from RLA errors, but provides a convenient way to recognize general classes of specimens.

The sequence number is usually an integer from 1 to 9999, and is unique within each accession number. Sequence numbers are assigned in the order in which specimens are sorted by year, location, and kind.

New sequence numbers may be created and interpolated into the existing series to correct sorting errors. These numbers, the only non-integral sequence numbers, are formed by concatenating a "/" and an integer "continuation number" to the sequence number preceding the one that should have been assigned. An example of this kind of sequence number might be 54/2, which would fall between 54/1 and 55.

In the almost 4000 specimens whose numbers are listed in Appendix G, there are 54 numbers to be interpolated, with a maximum of 11 between one pair of original sequence numbers.

2.4.3.3 Counting and weighing

The number of pieces in each specimen or subspecimen is counted or computed by a technique chosen to fit the kind of specimen and the number of pieces in it. In some cases, the containers holding the specimens may be counted instead of pieces.

Some specimens and subspecimens are weighed on a postal scale. Most weights are between 0.5 and 80 ounces (5 pounds), and are expressed to the nearest ounce or half-ounce. Weights outside this range are commonly expressed as "< 0.5 oz." or "> 5 lbs."

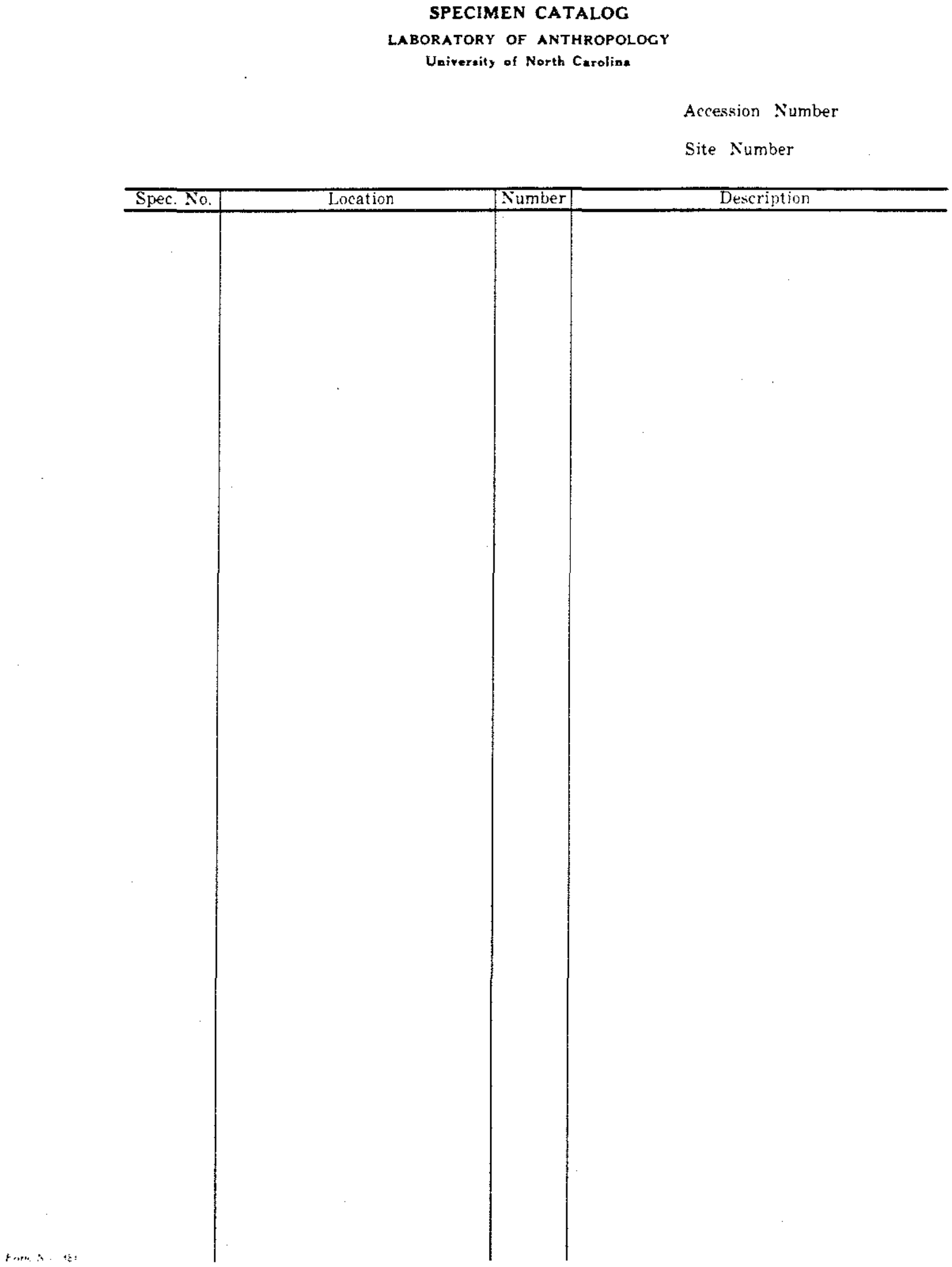
2.4.3.4 Recording

Specimen-catalog entries are made when specimens are recorded on Specimen Catalog forms (Figure 1). The information from each specimen-catalog entry is also written on the paper bag in which the specimen is placed for permanent storage.

The information given at the top of each Specimen Catalog form always includes an accession number. If all specimens with the accession number are from the same site, the site number, and perhaps the field season during which the specimens were collected, are also written at the top of the form.

The body of the form is used for entries for individual specimens. Each entry has a value in each column of the form. A value too long to fit into its column is continued on succeeding lines of the same column. Ditto marks (") are used extensively to indicate that a specimen has all or part of a value in common with its predecessor on the form.

The "Spec. No." column is used for the specimen type code and sequence number. All information about where the specimen was collected (including the site number, if it was not recorded at the top of the form) is entered in the "location" column. The specimen's count, including the units counted if they were not individual pieces, is entered in the "Number" column. The entry in the "Description" column is a description of the kind of specimen. If the



SPECIMEN CATALOG
LABORATORY OF ANTHROPOLOGY
University of North Carolina

Accession Number

Site Number

Spec. No.	Location	Number	Description

Form N [illegible]

Figure 1: Specimen Catalog form
(Reduced from 8 1/2 x 11 inches)

specimen was weighed, its weight may be entered in either the "Number" or the "Description" column.

Subdivided specimens, which have several counts and may have several weights, are entered in one of two ways. In the first, the "Number" column contains the total count and weight for the specimen, and the "Description" column contains a general specimen description and the counts, weights, and descriptions for the subspecimens. The second way, which has no general description, total count, or total weight, puts subspecimen counts and weights in the "Number" column and subspecimen descriptions in the "Description" column.

2.5 DATA CHARACTERISTICS

Data in the specimen catalog is characterized by variability and fixedness. The variability results from differences in recording formats and in information. Recording formats vary because of differences between archaeologists; information varies because of differences in specimens or collection locations. Although variability in recording formats can be eliminated without loss of information about anything except recording techniques, variability in information cannot. Unfortunately, the distinction between these sources of variability is so subtle that in some cases it can be drawn reliably only by an RLA archaeologist familiar with the sites from which the specimens were collected.

The variability of the data also makes detecting errors in the specimen catalog difficult. Error detection requires distinguishing data that is wrong from data that is merely about an idiosyncratic specimen. Reliably making this distinction for location information requires an RLA archaeologist familiar with the location; making it for the specimen's description, count, or weight requires correct observations for comparison with the suspected errors.

The "fixedness" of the data refers to its being left unchanged from the time it is first recorded. In part, the fixedness arises because only the results of the preliminary processing associated with specimen definition and cataloguing are entered in the specimen catalog. Another reason for the fixedness is that, as it is observations of unique specimens rather than repeated observations of the same ones, the data is never superceded.

Yet another reason for the fixedness, more important for understanding problems in converting data to a standard format, is the difficulty of changing data. Because the specimen catalog is in Chapel Hill and the specimens are in Durham, changing the content of the specimen catalog either introduces discrepancies between the specimen catalog and the specimen bags or requires a trip to Durham to retrieve the affected specimens and change their bags.

Chapter III

SYSTEM DESIGN

This chapter, the last in the introductory section of this paper, describes the expected usage patterns for SPECS and design priorities set by the RLA, summarizes SPECS, and specifies a programming language and operating system in which SPECS could be implemented.

3.1 USAGE PATTERNS

The expected uses of SPECS have the following characteristics:

1. Fast turnaround is not essential. One-day turnaround is certainly acceptable, and turnaround of several weeks is often adequate. If the users need access to data that has not yet been entered, they can either wait for the data to be entered or temporarily revert to manual retrieval techniques and the original specimen catalog.
2. SPECS will be used by a small group of archaeologists from the RLA, who expect to enter their own data.
3. The need for information about each specimen depends on the specimen's pertinence to current research. Some specimens may never be of enough interest to justify entering information about them.

4. Once it is entered correctly, data is unlikely to change.
5. It is unlikely that information about a specimen will be deleted from the specimen catalog.
6. Although the data must be current when it is retrieved, no regular cycles are expected for data entry, modification, or retrieval.

3.2 USER-IMPOSED PRIORITIES

Dr. Trawick Ward and Mr. Jack Wilson, designated by the Director of the RLA as spokesmen for the RLA in SPECS design issues, were interviewed in October, 1980, about characteristics they thought were important in a computerized specimen-catalog data system. Their responses indicated two main concerns:

1. The system should be cheap to operate.
2. The system should protect the privacy, quality, and security of the data. More specifically, the data should be protected from damage or loss because of failure in the system or its operating environment, from access by unauthorized users, from unintended major damage caused by authorized users, and from entry of erroneous data.

Dr. Ward and Mr. Wilson considered flexibility (as exemplified by the ease of modifying SPECS to accept new kinds of data), response time, and convenience of use (as shown by a simple instruction set, the ability to define "procedures" of SPECS instructions, and a SPECS facility for writing instructions for packaged programs to access the data) as less important. They volunteered to learn PL/I or use packaged programs if it were required by SPECS design.

3.3 OVERVIEW OF SPECS

SPECS is a computerized system that provides a standard machine-readable format for data from the RLA specimen catalog. All of the data in the present specimen catalog can be entered in SPECS, or the users may enter only data for which they want computerized retrieval. The amount of data stored in SPECS will continue to grow, at a rate determined by the users, throughout the life of the system.

Data in SPECS is stored in a tape-based "master file," where it is periodically updated. A disk-based "transaction file" stores information about changes to be made to the master file. The changes requested on the transaction file may add new specimens to the master file or correct errors in data already entered. These changes are made during a "batch update," which produces both a new version of the master file and a new transaction file for changes to be made to the new version of the master file. Changes that have been requested do not affect the current version of the master file, but are incorporated into the new version produced by the next batch update.

SPECS also maintains a "site-yr-acc" file, in which it stores information about the relationships between sites, years, and accession numbers, and a "system parameters" file, in which it stores passwords and other bookkeeping information.

SPECS performs three basic kinds of activities: interactive sessions, batch updates, and data retrieval. These activities occur at different times, and serve different purposes.

Interactive sessions, in which the user communicates with SPECS by typing at a terminal, build the transaction file by collecting requests for master-file changes. During interactive sessions, a user can review or change information in the site-yr-acc and system parameters files and request that the master file be updated. SPECS allows one user during an interactive session. At the end of each session, SPECS prints a "session report" to tell the user about the results of the session.

Specimen-catalog data is entered into SPECS by typing it at a terminal during an interactive session. The format for data entry is enough like the Specimen Catalog forms that the data can be entered without intermediate re-encoding, but differs in that subspecimens may be represented as separate specimens. Appendix A describes data-entry formats.

Batch updates use the current master file to create new master and transaction files. They are done by a batch program (that is, one with which the user cannot communicate). Batch updates are performed after the interactive sessions in which the user requests them, or when SPECS requires them because the transaction file has grown too large. An "update report" is printed after each batch update to tell the user about the results of the update.

Data is retrieved by reading the master file and copying from it the information that is of interest in a particular situation. SPECS does not itself provide techniques for data retrieval. Instead, the user is expected to use SAS [references 3, 5, 6, 7] or a similar packaged program, to read the master file. SAS can print data in a format specified by the user, copy it to another machine-readable file, or use it for statistical analysis.

SPECS protects its data from three sources of damage: unauthorized users, authorized users, and the operating environment.

Unauthorized users damage the data when they read or change any of the SPECS files. During interactive sessions, SPECS protects its files by refusing to change any file or show its contents to a user until the user has given the correct password. The operating environment protects SPECS files at all other times. Chapter 8 describes protection from unauthorized users, and ways of allowing different kinds of use by different people.

Authorized users can damage the data by using the wrong master and transaction files during an interactive session or batch update. As described in Chapter 9, SPECS refuses to run interactive sessions and batch updates unless it is given the newest versions of the files. Although authorized users can tell SPECS to use other versions, its insistence on using the files it expects helps keep the users from unintentionally updating the wrong file.

The data can also be damaged by authorized users who accidentally give SPECS erroneous information. SPECS protects itself by using three kinds of validation for specimen-catalog data: single-field validation, multiple-field validation, and sequence checking. Single-field validation ensures that the data is correctly formatted and that it meets the other criteria specified in Appendix C. Multiple-field validation confirms that the site, year, and accession number, and specimen type, description, count, and weight, are consistent for a single specimen. Sequence checking examines pairs of specimens with sequential specimen numbers. Most multiple-field validation, and some single-field validation, uses "validation constants" set by the users and stored in the system parameters file. Some multiple-field validation also uses the site-yr-acc file.

As discussed in Chapters 5 and 6, SPECS' reactions to errors found by validation differ in interactive sessions and batch updates. SPECS allows the user to overrule it when it detects errors, recognizing that authorized users are the ultimate authorities on data correctness.

The operating environment can make SPECS files unreadable or lose them. SPECS protects itself against such damage by maintaining extensive backup files and providing techniques for using them. As described in Chapter 9, any current SPECS file can be reconstructed.

3.4 LANGUAGES AND OPERATING ENVIRONMENT

The SPECS programs will probably be written in PL/I [reference 2], a language that allows easy file handling, writes data in formats that can be read by SAS, and is widely used at UNC-CH. SPECS is designed to run under some version of TSO (IBM'S Time-Sharing Option) at TUCC (the Triangle Universities Computation Center, in Research Triangle Park, North Carolina). These choices of language and operating environment may be modified when the system is implemented.

Although SAS will allow the users to read the SPECS files, it lacks the ability to decide whether one character string is contained in another (whether "sherd" is contained in "potsherds," for example). This ability would so much help the users to recognize specimens in which they are interested that a SPECS implementation includes a SAS subroutine or function to recognize contained character strings. SAS subroutines may be written in PL/I or FORTRAN, but SAS functions for character-processing must be written in assembler [reference 5: 17, 108].

Chapter IV

FILES

4.1 INTRODUCTION

This chapter describes the files that SPECS uses for data entry and storage. Backup copies of some of these files may be kept, as described in Chapter 9. Other files, used for data retrieval, are presented in Chapter 7.

There are six files used for data entry and storage:

1. The master file, which contains information about specimen-catalog entries. Master files are updated and rewritten during batch updates.
2. The transaction file, which contains requests to modify the master file. A transaction file is begun by a batch update, used to accumulate transactions requested during interactive sessions, and read during the next batch update.
3. The site-yr-acc file, which contains information used for validation of the relationships between sites, years, and accession numbers. It is read and written during interactive sessions and read during batch updates.
4. The system parameters file, which contains information that SPECS needs to remember for its

operation. This information includes the SPECS passwords, used only during interactive sessions, validation constants, used during batch updates and interactive sessions, and information for recognizing master and transaction files and for assigning transaction numbers.

5. The session report file, which holds session-report entries during an interactive session.
6. The update report file, which holds update-report entries during a batch update.

Table 2 enumerates the records in each file. Appendix B contains detailed descriptions of the fields in the records of all but the report files, and Appendix D enumerates possible report entries.

TABLE 2

Files and records

File	Records
Master	Master-file header Specimen-data
Transaction	Transaction-file header Insert-data Delete-data Change-data Cancel-transaction Validation-override Validation-restoration
Site-yr-acc	Site-yr-acc
System parameters	Passwords Validation constants Progress
Session report	Session-report entry
Update report	Update-report entry

4.2 THE MASTER FILE

4.2.1 Records

Each master file contains one master-file header record and zero or more specimen-data records. SPECS uses the master-file header record to recognize master files and distinguish master-file versions. The header record for each version of the master file contains information about when, and with what master and transaction files, the version was created.

Each specimen-data record contains information about one specimen or subspecimen.

4.2.2 Key structure

Master-file keys, which represent specimen numbers, are composed of three parts. The first part, a 4-digit integer, is an accession number. The second, a 6-digit integer, is formed by concatenating a 4-digit sequence number and a 2-digit continuation number. The third part is a one-character abbreviation of the specimen type code. Section 1.8 of Appendix C specifies how these parts are extracted from a specimen number.

Two peculiarities of this choice of keys need explaining:

1. Although continuation numbers are needed to uniquely distinguish only a small fraction of the specimens (54 of the almost 4000 specimens listed in Appendix G), they are represented for all specimens to allow the master file to have a fixed-length key representing a specimen number.
2. Because the combination of accession number and sequence number (including its continuation, if any) is unique for specimens with unique specimen numbers, the specimen type should not influence the order into which master-file records are sorted. The specimen type is included in the key to help prevent typing errors in specimen numbers from causing transactions with mistyped specimen numbers to be applied to unexpected records. The specimen type also distinguishes between specimens of different kinds to

which the RLA has erroneously given the same accession, sequence, and continuation numbers.

4.2.3 <u>Non-unique keys</u>

Because SPECS can treat subdivided specimens as several specimens with the same specimen number, specimen-data records for the subspecimens may have the same master-file keys. The records for these subspecimens will also have the same location information, and may coincidentally have the same count and weight, but can be distinguished by their descriptions.

It is difficult to predict the number of specimen-data records that may have the same key, because this depends on an RLA policy decision about how subspecimens are entered (see Appendix A). The specimen-catalog entries studied for this paper show nine kinds of beads or five kinds of potsherds for one specimen, but suggest that specimens with more than two or three subspecimens are rare. Subspecimens are recorded in 91 of the 1417 specimen catalog entries from SkV1a that are listed in Appendix G, but there are none among specimens from the other sites.

4.2.4 <u>Sorting order</u>

The master file is sorted into increasing numerical order of accession, sequence, and continuation numbers, then into alphabetical order by specimen type. As was mentioned in the description of master-file keys, the specimen type should not affect the sorting order if specimen numbers were assigned correctly. The sorting order of records with the same keys is undefined, although the batch update maintains them in the order in which the specimens were entered.

Because of its key value (specified in Appendix B), the master-file header record is always the first record in the file.

4.2.5 File size, growth, and activity ratio

The size and growth of the master file are difficult to predict, being dependent how much data the RLA enters. If the file has 1,000,000 records at an average of 50 bytes each, it will occupy fifty megabytes. New specimens are added to the specimen catalog at a rate of about 5,000 per year. Most of these specimens are likely to be of interest in current research, so they will probably be entered promptly.

The activity ratio of the master file is also difficult to predict. For batch updates, it is likely to be very low. Records used during a batch update will be either new or erroneous, since correctly entered records do not need to be updated. If the specimens are entered in specimen-number order, and if most errors are corrected during the first batch update after they are incorporated in the master file, activity during batch updates will be clustered at the end of the file.

The activity ratio for reports is likely to be higher than for batch updates, and the records that are needed may be more widely scattered. The number of records actually used in a report depends on the report, but it will probably be necessary to examine all of the specimen-data records to find the needed ones.

4.3 THE TRANSACTION FILE

4.3.1 Records

The transaction file has one or more transaction-file headers and zero or more records of other kinds. Each transaction-file header record, of which there is normally one, identifies a transaction file and tells SPECS which master file it is intended to update.

The other records in the transaction file, collectively known as transactions, may be either data-modifying or transaction-modifying. Each data-modifying record contains a request to change one or more specimen-data records with a common key. Each transaction-modifying record affects the way SPECS treats a single data-modifying or transaction-modifying transaction during a batch update.

The three kinds of data-modifying transactions are:

1. Insert-data records, which provide data for new specimen-data records.
2. Delete-data records, which request that one or more specimen-data records be deleted from the master file.

3. Change-data records, which provide information to replace an earlier value in one or more specimen-data records.

Delete-data and change-data records are rare, since both are needed only to correct errors. Indeed, the most common use for a delete-data record may be to remove a specimen-data record so erroneous that it is easier to enter the record again than to correct it.

The three kinds of transaction-modifying records are:

1. Validation-override records, which tell SPECS to execute an insert-data or change-data transaction even if the new data in it fails validation during a batch update.
2. Validation-restoration records, which tell SPECS to refuse to execute an insert-data or change-data transaction if the new data in it fails validation during a batch update. By default, changes that fail validation are not made; thus, a validation-restoration record has no effect unless there is a previous validation-override record for the same insert-data or change-data transaction.
3. Cancel-transaction records, which request that a previously entered transaction should not be executed during a batch update.

4.3.2 Key structure

The first of the two keys used in the transaction file is a specimen number, in a representation identical to that of the master-file key. In data-modifying transactions, the specimen number is the master-file key of the specimen-data records to be modified. In transaction-modifying records, the specimen number is the master-file key of the specimen-data record affected by the transaction to be modified. If, for example, there is a request to change information in a specimen-data record with a master-file key 4213051700a, the change-data record and any transaction-modifying records affecting it will all have master-file keys of 4213051700a.

The second transaction-file key is the transaction number, an integer uniquely assigned to each record when it is written on the transaction file.

4.3.3 Sorting order

The transaction file is generated in transaction-number order, then sorted by specimen and transaction numbers at the beginning of each batch update. Sorting by specimen number groups the transactions that affect each specimen and puts them into master-file order. Sorting by transaction number then puts the transactions for each specimen number into the order in which they were requested by the user.

4.3.4 File size

The size of the transaction file at any time is hard to predict because it depends on the user-determined rate of data entry and frequency of batch updates. The file size is bounded by the available space; SPECS forces a batch update when the file outgrows its maximum size. The maximum size for the file is chosen by the implementor to minimize the total costs of unneeded disk space and forced updates.

4.4 THE SITE-YR-ACC FILE

4.4.1 Records, keys, and file access

The site-yr-acc file contains zero or more site-yr-acc records, each indicating a site, a year, and an accession number assigned to specimens from that site and year.

The site and year are concatenated to form the file key. Key values are unique unless more than one accession number was assigned to specimens from the same site and year. (Multiple accession numbers for a single site and year are rare or nonexistent, and can only result from an RLA error in assigning accession numbers. Nevertheless, SPECS handles them, rather than forcing the user to correct them.)

Access to the site-yr-acc file is direct, with the block in which each record is stored being calculated by a hashing function. The records within each block are sorted by their keys, so that records with the same key can be readily found. Sorting the blocks in each record also means that a binary search can be used to find individual records.

IBM does not support blocked direct-access files [reference 1: 9-6,9-13 to 9-14], so SPECS must block and deblock the site-yr-acc file.

4.4.2 File use

The expected uses of the site-yr-acc file fall into two groups: data-file building and validation.

Data-file building occurs when information about sites, years, and accession numbers is entered into the site-yr-acc file without entering information about specimens from the site and year in the master file. It may be used to provide data for a report about sites, years, and accession numbers for which the RLA is not interested in individual specimens.

While the file is being built, many records are entered but few are read or changed. The new entries are likely to be sorted by accession number, and may be at least partially sorted by site and year.

After the file is built, many records will be read. If SAS reads the file, it will use a sequential pass through the entire file [reference 7: 68].

The vast majority of file use will probably be for validation. In validation, records are read, and perhaps written, every time the user gives an accession number for a new specimen during an interactive session or changes a site, year, or specimen number during a batch update. The relative frequencies of reading and writing in the file are impossible to predict, since they depend on the data-entry styles of individual users. The absolute frequency of file accesses may also vary among data provided by different users, but is likely to be higher for specimens from surveys than for those from excavated sites because of the surveys' lower average numbers of specimens per site.

4.4.3 File size

The size of the site-yr-acc file is difficult to predict. The file may have one record for every combination of sites and years for which the RLA has specimens. Data-file building will, of course, bring the file closer to this size than using it exclusively for validation.

There is no obvious source of exact information about the number of combinations of sites and years represented in the RLA collections, although it is unlikely that there are more than about 11,500. The RLA has records tor 8000 to 10,000 sites, the vast majority of which are visited during only one year. Perhaps twenty sites have been visited annually for fifty years. Excavations have been made at about ten sites a year for forty years.

4.5 THE SYSTEM PARAMETERS FILE

4.5.1 Records

The system parameters file has 150 bytes of information in three records. The length of each record is constant, but no two records are the same length. The records are:

1. A password record, containing all SPECS passwords except the one needed to restore the system parameters file (see Chapters 8 and 9).
2. A validation-constants record, containing the interactively changeable values used during some single-field and multiple-field validation.
3. A progress record, containing information needed to identify the current master and transaction files and assign new transaction numbers.

4.5.2 Key structure and record position

Records in the system parameters file are accessed directly. Each record has a unique one-digit key representing its relative position in the file.

4.6 THE SESSION AND UPDATE REPORT FILES

Session and update report files are allocated at the beginning of each interactive session or batch update, used to collect all report entries generated during the session or update, printed, and deleted.

Chapter V

INTERACTIVE SESSIONS

Interactive sessions are dialogues between SPECS and the user, in which SPECS repeatedly asks for information, receives it, and acts upon it. These dialogues, and the output they produce, are specified in Appendix E, which should be consulted to answer any questions of detail about interactive sessions.

SPECS can perform six kinds of activities during interactive sessions:

1. Reconstructing the system parameters file.
2. Resetting constants used for master-file reconstruction.
3. Command interpretation.
4. Validation and conversion of data from the format in which the user enters it to the format in which SPECS stores it.
5. Scheduling batch updates.
6. Generating session reports.

5.1 UNIVERSAL RESPONSES

SPECS accepts the "universal responses" ":", "?", and "Q" whenever it wants information, and interprets them as follows:

1. ":" is a request for a menu listing the responses SPECS will recognize and the interpretations it will make of them.
2. "?" is a request for an elaboration of SPECS' request for information, and perhaps an explication of the consequences of various responses.
3. "Q" (short for "quit") asks SPECS to stop whatever it is doing. The interpretation of the "Q" command depends on the situation in which the user gives it, but is always described in the current menu. SPECS acknowledges a "Q" command with a message telling the user what it has stopped.

5.2 RECONSTRUCTING THE SYSTEM PARAMETERS FILE

At the beginning of each interactive session, SPECS tries to read the system parameters file. If the file cannot be read, SPECS reconstructs it by asking the user for each of the values that was lost.

5.3 RESETTING CONSTANTS FOR MASTER-FILE RECONSTRUCTION

After the system parameters file is read or reconstructed, SPECS asks the user whether he would rather reset the constants used to reconstruct the master file or give SPECS commands to execute. If the user says he wants to set the constants, SPECS asks him for the file identification and the next transaction number.

5.4 COMMAND INTERPRETATION

If the user does not want to reset the constants for master-file reconstruction, he can give SPECS the commands listed in Appendix A. SPECS repeatedly asks the user for a command, confirms that he is authorized to give the command, and executes it. It recognizes the END command as indicating that the user has finished giving commands.

Command interpretation serves five purposes:

1. Ensuring that SPECS accesses files only for users who are authorized to command it to.
2. Asking the user for instructions for the next batch update, and writing transaction-file records to ensure that the instructions are followed.
3. Allowing the user to interactively change the information in the site-yr-acc file.
4. Allowing the user to reset the validation constants.
5. Writing the user's comments on the session report.

5.4.1 Ensuring that users are authorized

When SPECS is given a command that requires a password that the user has not given earlier in the session, it asks the user for the password. SPECS does not execute the command until the password is given.

5.4.2 Writing transaction-file records

SPECS recognizes six commands that tell it to write records on the transaction file: CANCEL, CHANGE, CHECK, CONFIRM, DELETE, and INPUT.

5.4.2.1 The CANCEL command

The CANCEL command tells SPECS to ignore a transaction during the next batch update. To execute a CANCEL command, SPECS must know the number of the transaction to be ignored and the number of the specimen it would affect.

When SPECS executes a CANCEL command, it writes a cancel-transaction record.

The session report entry for a CANCEL command includes the number of the transaction to be cancelled, the number of the specimen it affected, and the transaction number of the cancel-transaction record.

5.4.2.2 The CHANGE command

The CHANGE command requests that data about one or more specimens with a single specimen number be replaced by a new value. The data that is changed may be in the master file or in the part of the transaction file that has already been written.

SPECS performs single-field validation for the new value, and writes messages produced by the validation at the user's terminal.

Each CHANGE command must be accompanied by the number of the specimens for which data is to be changed, the kind of information to be changed, and the new value. The command may also include instructions about which records are to be changed if several with the same specimen number are found.

The three ways to instruct SPECS about which specimens with the same number should be changed are:

1. Telling it that all of the specimens should be affected. In this case, all specimens with the appropriate specimen number (including any entered by earlier transactions in the current file) will be changed.
2. Giving a substring to be matched in the descriptions of the specimens to be changed. In this case, only records for specimens whose descriptions include the substring will be affected. For records already in the master file, the description that is used is the one from the beginning of the batch update; for newly entered specimens, SPECS uses the description with which the specimen was entered.

3. Doing nothing. In this case, SPECS assumes that there is only one record for the specimen number. If it finds more than one during a batch update, SPECS prints a message on the update report and ignores the request to change the data.

In executing the CHANGE command, SPECS writes a change-data record on the transaction file.

The session report entry for a CHANGE command includes the number of the specimen to be changed, the kind of information to be changed, any instructions for processing multiple specimens with the same specimen number, the new value as the user entered it, the new value as it is represented in the SPECS files, messages produced during validation of the new value, and the transaction number of the change-data record.

5.4.2.3 The CHECK command

The CHECK command tells SPECS not to execute an insert-data or change-data transaction in the current transaction file if the data given by the transaction fails validation during the next batch update. Because this is the default treatment for transactions giving data that fails validation, a CHECK command is needed only to counteract the effect of a previous CONFIRM command.

With each CHECK command, the user must tell SPECS the number of the transaction giving the new data and the number of the specimen for which the data is provided.

SPECS writes a validation-restoration record on the transaction file when it executes a CHECK command.

The session report entry for a CHECK command includes the number of the transaction giving the new data, the number of the specimen for which the data is given, and the transaction number of the validation-restoration record.

5.4.2.4 The CONFIRM command

The CONFIRM command confirms that the data supplied by an insert-data or change-data transaction in the current transaction file is correct, and requests that the transaction be executed even if the data it gives fails validation during the next batch update. SPECS prints any validation error messages for the new data during the batch update, but then considers the data correct.

In a CONFIRM command, the user must tell SPECS the number of the transaction giving the new data and the number of the specimen for which the data is provided.

SPECS writes a validation-override record on the transaction file when it executes a CONFIRM command.

The session report entry for a CONFIRM command includes the number of the transaction giving the new data, the number of the specimen for which the data is given, and the transaction number of the validation-override record.

5.4.2.5 The DELETE command

The DELETE command tells SPECS to remove one or more records with a common specimen number from the master file. A DELETE command may also delete insert-data records already written in the transaction file.

Each DELETE command must include the number of the specimen for which the information is to be deleted. It may also include instructions about which records are to be deleted if several with the same specimen number are found, using the same conventions as in the CHANGE command.

When SPECS executes a DELETE command, it writes a delete-data record on the transaction file.

The session report entry for a DELETE command includes the number of the specimen to be deleted, any instructions for processing multiple records with the same specimen number, and the transaction number of the delete-data record.

5.4.2.6 The INPUT command

The INPUT command prepares SPECS for information about new specimens. SPECS repeatedly asks the user for data and writes insert-data records about it on the transaction file. To stop the INPUT command, the user enters a "Q" when SPECS asks for information about a new specimen.

When SPECS asks for information about a new specimen, it recognizes three kinds of response:

1. the universal responses ":", "?", and "Q", which are given their usual interpretations.
2. "HEADER", which tells SPECS to print column headings for the input at the user's terminal.
3. anything else, which is interpreted as information about a new specimen.

To convert information about a new specimen to an insert-data record, SPECS performs four steps: identifying information, validation, editing values, and writing.

Appendix A describes the input format used for new specimens and the techniques SPECS uses to identify the information it is given. These techniques are specified in Appendix E.

SPECS applies both single-field and multiple-field validation algorithms to data about a new specimen. If the specimen is not the first to be entered after an INPUT command, SPECS also uses sequence checking. All of these algorithms are specified in Appendix C.

If, after seeing the error messages produced during validation and the values SPECS will write in the insert-data record for the new specimen, the user wants to change some of the information, he may edit as much as he wants to. SPECS repeats the single-field validation, and perhaps some of the multiple-field validation and sequence checking, for the changed information.

The session-report entry for each insert-data record includes the information that the user gave, SPECS' representation it, messages produced by validation for the most recent values of the input fields, and the transaction number of the insert-data record. Validation messages produced for replaced values are not printed.

5.4.3 Accessing the site-yr-acc file

SPECS accesses the site-yr-acc file when it receives a SET-ACC or SHOW-ACC command from an authorized user and when information about sites, years, and accession numbers is given with an INPUT command.

5.4.3.1 The SET-ACC command

The SET-ACC command adds an entry to the site-yr-acc file or deletes all entries for a site and year. To execute the command, SPECS needs to know a site, a year, and either an accession number or that there is none.

SPECS takes one of three actions for a SET-ACC command:

1. If there is no accession number, SPECS deletes any site-yr-acc records it finds for the site and year.
2. If there is no record for the site and year, SPECS performs single-field validation for the site, year, and accession number. If the validation fails, SPECS refuses to execute the SET-ACC command. If the validation succeeds, SPECS writes a new site-yr-acc record.

3. If there are already records for the site and year, SPECS prints their accession numbers at the user's terminal, then asks whether the new number replaces the old ones. If the user says it does, SPECS deletes the records with the old accession numbers, then writes a record with the site, year, and new accession number; otherwise, SPECS simply adds the new accession number to the file.

The session report entry for a SET-ACC command includes the site, year, and accession number, and any accession numbers already in the file for that site and year. If SPECS deletes any records, the report lists their accession numbers.

5.4.3.2 The SHOW-ACC command

The SHOW-ACC command reads the site-yr-acc file to show the user any accession numbers for the site and year he specifies. The session report entry for a SHOW-ACC command gives the site, year, and the accession numbers recorded for them.

5.4.4 Resetting passwords and validation constants

The RESET command allows authorized users to reset a password or validation constant in the system parameters file. SPECS shows the user the current value of the password or constant, prompts him for a new value, and writes it in the file.

The new values of validation constants replace the old values when the RESET command is executed. SPECS recognizes the new values of passwords, but does not require the user to give new passwords if he already gave the old ones.

5.4.5 Writing comments on the session report

The COMMENT command tells SPECS to write a comment on the session report. The session report entry for a COMMENT command is "USER'S COMMENT:" followed by the comment.

5.5 VALIDATION AND FORMAT CONVERSION

During interactive sessions, SPECS uses validation to check the correctness of specimen-catalog data and convert it from the format in which it is entered to the formats used in master-file and transaction-file records. SPECS uses single-field validation for the data given with INPUT or CHANGE commands, all specimen numbers, and the sites, years, and accession numbers given with SET-ACC and SHOW-ACC commands. SPECS uses multiple-field validation and sequence checking for data given with the INPUT command.

The validation algorithms are specified in Appendix C. This chapter limits itself to describing SPECS' actions for errors found during interactive sessions.

5.5.1 Single-field validation

Errors found during single-field validation provoke one of four responses:

1. SPECS may ask the user to repeat information. If SPECS cannot tell what kind of grid-square information was given, for example, it asks the user to re-enter the grid-square data.
2. SPECS may approximate values so they can be represented, and may store the original information in another field of the master-file or transaction-file record. If the user objects, he can tailor the data so that SPECS will not need to approximate it, or edit the value of the field in which SPECS puts the original information.

   Four examples of data that SPECS approximates are:

   a) Burials represented by letters and numbers (e.g. Burial 49A). In this case, the information is approximated as Burial 49. The original information, abbreviated as B 49A, is represented as miscellaneous location information.

   b) Point proveniences. In this case, the location is approximated by the coordinates of the southeast corner of the one-foot square in which the point provenience lies. The original point provenience data is represented as miscellaneous location information.

   c) Specimen descriptions that are longer than 127 characters are approximated by their first 127 characters; any information beyond the 127'th character is lost.
   d) Counts that are greater than 32,767 are approximated as 32,767.
3. SPECS prints error messages about data that can be represented or approximated but appears erroneous. The user may change the data if he wishes; if he does not, SPECS assumes that the information is correct.
4. SPECS refuses to accept data that it cannot represent or approximate. If the data is given with an INPUT command, SPECS asks the user for new information; otherwise, it refuses to execute the command with which the data was given.

5.5.2 Multiple-field validation and sequence checking

When information about a new specimen includes a site, year, and accession number that SPECS does not recognize as consistent, SPECS refuses to proceed until the user either confirms that the values are consistent or gives a site, year, and accession number that seem correct.

SPECS prints error messages about specimen descriptions, types, counts, and weights that seem inconsistent, but does not force the user to change them. Similarly, it tells the user about information that seems out of sequence, but accepts the data if he does not change it.

5.6 SCHEDULING BATCH UPDATES

At the end of every interactive session, SPECS asks the user whether to run a batch update. If the user tells it to, and gives the correct password, SPECS tells the operating environment to run the update.

SPECS requires a batch update when the transaction file is full.

5.7 SESSION REPORTS

Session reports summarize interactive sessions. The information on each session report is dictated by the user's actions, the logic detailed in Appendix E, and the possible report entries in Appendix D.

A session report has three parts: introduction, summary of processing, and listing of system parameters. The introduction identifies the report as a SPECS session report, and gives the date, the identification of the current transaction file, and the user's name. The summary of processing describes either information that was set for master-file reconstruction or SPECS' responses to the user's commands. The listing of system parameters contains the current validation constants and progress information and, if they are requested by an authorized user, the current passwords.

Chapter VI

BATCH UPDATES

6.1 INTRODUCTION

Batch updates make the master-file changes requested during interactive sessions. Each update produces a new version of the master file, a new transaction file, and an update report. The new master and transaction files become the current versions, to be used for interactive sessions and reports until they are replaced.

During a batch update, SPECS uses a single pass through the current master and transaction files. In the pass, it reads all master-file and transaction-file records with the same master-file key as a group. When SPECS processes the records in a group, it uses insert-data transactions to create new specimen-data records, change-data transactions to modify specimen-data records from the master file and earlier insert-data records, and delete-data transactions to prevent specimen-data records from being written in the updated master file. Specimen-data records in groups with no transaction-file records are simply copied from the current to the new master files.

The transaction-modifying cancel-transaction, validation-override, and validation-restoration records change the effects of other transactions. SPECS processes cancel-transaction records by ignoring the transaction that is cancelled. Validation-override records cause SPECS to accept change-data and insert-data transactions as correct even if they give data that fails batch-update validation. Validation-restoration records tell SPECS to refuse data given by insert-data and change-data transactions if it fails validation. SPECS ignores all but the most recent uncancelled cancel-transaction, validation-override, or validation-restoration record for a single transaction.

Two principles govern batch updates:

1. Specimen-data records represent separate specimens, even if they have the same key value. Thus, the applicability of a transaction that has instructions for multiple records with the same specimen number, and the results of applying the transaction, may differ between master-file records with the same key.
2. No transaction can be influenced by transactions that were requested before it was. Thus, a transaction cannot be cancelled, or have its validation overridden or restored, by an earlier cancel-transaction, validation-override, or validation-restoration record. Further, specimen-data records created from insert-data transactions are never affected by earlier change-data or delete-data records.

Sequence-checking validation is the sole exception to this rule. As described in section 6.2, the processing of a change-data or insert-data record may be influenced by comparing it with data from earlier change-data or insert-data records.

SPECS uses two batch updates to change master-file key values for records that are already in the master file. In the first update, SPECS makes any other changes requested for the records and performs multiple-field validation for the new values. If the validation fails and is not overridden, SPECS refuses to change the master-file keys.

If validation succeeds for the records with the changed keys, SPECS deletes the records from the master-file position established by their old key values and writes insert-data records on the new transaction file. The insert-data records, which contain the information from the records with the changed keys, add the changed records in the correct master-file positions for their new keys during the second batch update.

6.2 VALIDATION

During batch updates, SPECS assumes that data that fails validation is erroneous unless the user has explicitly confirmed its correctness. This assumption ensures that the user has seen all error messages for the data before it is considered correct (unless, of course, he has confirmed it in advance of the batch update, in which case the error messages accompany the acceptance of the data).

Batch-update validation avoids repeating validation done during interactive sessions and earlier updates. It is limited to the following situations:

1. Multiple-field validation checks the consistency of values provided by change-data records. The values to which the changed information is compared may be from an insert-data record, one or more change-data records, an existing specimen-data record, or some combination of these.
2. Sequence checking examines pairs of specimen-data records. It is used when location information is changed and when the first specimen in a pair is from the master file and the second is from an insert-data record. Sequence checking compares specimens with different numbers.

SPECS uses validation for the final values of each specimen-data record. Data given by cancelled transactions and values that are later changed are not checked. The order in which the values are set does not affect their validation.

If the final values for some of the fields in a specimen-data record are inconsistent, all of the transactions that set the inconsistent values are considered erroneous when applied to that specimen-data record. SPECS treats transactions that set erroneous values in three ways:

1. If validation is overridden for the transaction, the value it sets is treated as though the it were correct.
2. If the transaction is a change-data record whose validation is not overridden, the fields set by the transaction retain their values from the beginning of the batch update.
3. If the transaction is an insert-data record whose validation is not overridden, SPECS refuses to write the specimen-data record created from it onto the new master file, even if all other transactions for the new record are correct or have overridden validation.

6.3 <u>THE UPDATE REPORT</u>

Each batch update prints an update report summarizing the changes made by the update. An update report has three sections: initialization, summary of processing, and shutdown. Update-report entries are listed in Appendix D.

The initialization section identifies the batch update and confirms that the expected master and transaction files were used. If the files were unusable, SPECS prints error messages in the initialization section of the report.

The processing section lists the transactions that were read and the specimen-data records that were added, changed, or deleted. It identifies cancelled transactions and transactions with overridden or restored validation, lists error messages, tells what was done about erroneous values, and lists master-file records.

The shutdown section includes counts of the specimen-data records that were treated in various ways and listings of current progress information and validation constants.

Chapter VII

DATA RETRIEVAL

Authorized users can use SAS [references 3, 5, 6, 7] to print reports about data from SPECS files. Each user may design his own reports, or the RLA may keep a library of SAS programs for standard reports. Reports that the RLA might want include:

1. A specimen-catalog facsimile. This report would list all information about all specimens in a given range of specimen numbers, in a format like that of a Specimen Catalog form.
2. A transaction report. This report would list all transactions in the current transaction file, in the order in which they would be processed by a batch update.
3. A report of sites, years, and accession numbers. This report, listing the accession numbers given to specimens collected from each site and year, could be generated from the site-yr-acc or master file. A report from the site-yr-acc file could include sites, years, and accession numbers for specimens that have not been entered in the master file; a report from the master file could include the numbers of the

specimens from each site and year. A site, year, and accession number report from the master file might be similar to Appendix G, but without the "Comments" column.

SAS can make machine-readable copies of the SPECS files. If the copies are in a different format than the originals, they may be used by other program packages like SPSS [reference 4], which cannot read the SPECS files. Analyzing SPECS' data with SPSS is probably most easily done by using SAS to write a new file with the desired data in a format that SPSS can read, then analyzing data from the new file.

Chapter VIII

PRIVACY

SPECS uses SPECS passwords, operating-environment passwords, and separate copies of files to allow the RLA to control who can use their data and request batch updates.

8.1 SPECS PASSWORDS

During interactive sessions, SPECS construes knowledge of the SPECS passwords as authorization to issue commands or provide data for file reconstruction. By restricting knowledge of the SPECS passwords, the RLA controls who can use SPECS, and for what.

There are ten SPECS passwords:

1. The parameters password, which authorizes the user to supply values for reconstructing the system parameters file.
2. The master password, which authorizes the user to change the SPECS passwords, reset the progress information for master-file reconstruction, and request that the SPECS passwords be printed on the session report.
3. The input password, which authorizes the user to issue an INPUT command.

4. The change password, which authorizes the user to issue a CHANGE command.
5. The delete password, which authorizes the user to issue a DELETE command.
6. The cancel password, which authorizes the user to issue a CANCEL command.
7. The validation-control password, which authorizes the user to issue a CONFIRM or CHECK command and to use a RESET command to change the validation constants.
8. The show-acc password, which authorizes the user to issue a SHOW-ACC command.
9. The set-acc password, which authorizes the user to issue a SET-ACC command.
10. The update password, which authorizes the user to request batch updates.

When SPECS needs confirmation that a user is authorized to issue a command or give information to recreate a file, it asks him for the correct password. The user is given three chances to provide the password; if he fails, SPECS prints a message on the session report and refuses to execute the command or accept the information. Each password must be supplied at most once during a session, since SPECS remembers that a user knows the passwords he has given.

The parameters password is encoded in a SPECS program, and can be changed only by changing the program. The other passwords, stored in the passwords record of the system parameters file, can be reset by using the RESET command and the master password or by recreating the system parameters file and supplying the parameters password.

8.2 OPERATING-ENVIRONMENT PASSWORDS

Operating-environment passwords keep unauthorized users from bypassing interactive sessions to read or change SPECS' files, and ensure that only authorized programmers can change the SPECS programs. Passwords are needed not only for the current master, transaction, system parameters, and site-yr-acc files, but also for all backup copies of these files.

Little can be said about the operating-environment passwords in the system design because they are so heavily dependent on the SPECS implementation. Nevertheless, the design requires that they be used to restrict file access in these ways:

1. Only people who have the RLA's permission to modify SPECS may read or change its programs.
2. Only SPECS may write in the system parameters, master, transaction, and site-yr-acc files.
3. Only SPECS programs and people authorized to retrieve data may read the transaction, master, and site-yr-acc files. Access to the files is controlled separately; for example, authority to read the master file does not convey permission to read the others.

4. Only SPECS programs and people authorized to know the SPECS passwords may read the system parameters file.
5. The ability to read and write all backup files is restricted to people authorized to read and write the files from which the backup files were made.

8.3 SEPARATE COPIES OF FILES

The RLA can also control access to SPECS data by using SAS to make separate copies of files. These copies, which may contain only part of the information currently entered in SPECS, are convenient to give to people to whom the RLA cannot, or will not, allow access to SPECS. Although the recipients of the copied files must be told how to read their copies, they need no other knowledge of SPECS or access to its files. By selecting the data to be written in each copy of the file, the RLA can control who can use what part of their data.

Copied files are useful for authorized users who want several reports or analyses of the same data. Analyzing a separate file of the data in which a user is interested is often cheaper than analyzing the entire master file, and guarantees that the data will not change when the master file is updated.

Copied files written on tape or punched on cards provide a convenient way to transfer data to users of computers other than the one on which SPECS is implemented.

Operating-environment passwords may be used to limit access to copies of files.

Chapter IX

SECURITY

SPECS protects its data from damage by the users and the operating environment. This protection includes the validation and prevention of access by unauthorized users discussed earlier, and also includes file-recognition techniques for the master and transaction files, backup copies or old versions of SPECS' files, and procedures for recreating any files that may be lost.

9.1 FILE-RECOGNITION TECHNIQUES

SPECS uses file-recognition techniques to ensure that transactions are written on the current transaction file and that the correct versions of the master and transaction files are used in batch updates.

9.1.1 Interactive sessions

During interactive sessions, SPECS detects three kinds of erroneous transaction files: files that are not really transaction files, transaction files that are not current, and "multiple-update" transaction files made by combining other transaction files. If it finds any of these, SPECS refuses to add new transactions to the file.

SPECS distinguishes transaction files from all other kinds of files by the "T" in the "specimen type" fields of their header records. In a current transaction file, the "transaction-file ID" field in the header record has the same value as the "file ID" field in the progress record of the system parameters file. Multiple-update transaction files have more than one header record.

9.1.2 Batch updates

File recognition during batch updates ensures that the master and transaction files are both contemporaneous and current. A transaction file is contemporaneous with the master file created during the same update, and contains the transactions intended to modify that master file. Current files have the current value of the "file ID" in the progress record of the system parameters file as their file identifications.

Batch-update file recognition imposes three restrictions on the files:

1. Each update uses one master and one transaction file.
2. The files in the update are contemporaneous.
3. The files in the update are current.

Multiple-update transaction files may be used during batch updates.

9.1.3 Protecting the information for file recognition

File recognition depends on information from three sources: master-file header records, transaction-file header records, and the progress record of the system parameters file. This information is protected from typographical errors by having it maintained as automatically as possible.

Values in the master-file and transaction-file header records are provided by SPECS during the batch update in which the files are created. These records are never rewritten.

When SPECS creates new master and transaction files, it updates the progress record of the system parameters file to show that the new files are current. The user should change the identification of the current files under only two circumstances: when he wants to revert to old master and transaction files and when the system parameters file is lost.

When the user provides new file identifications, SPECS insists that they must be lower than the next transaction number to be assigned. Although this insistence does not guarantee that the identifications refer to existing file versions, it ensures that the file identifications cannot refer to files that have not yet been created.

9.2 BACKUP FILES AND OLD VERSIONS

Backup and outdated copies of files help the user reconstruct the current versions of SPECS files if the original copies are lost. Backup files are kept for the transaction and site-yr-acc files, and for all files containing SPECS programs. No separate backup copies of the master files are made; instead, old versions of master and transaction files are kept. Because the system parameters file is small and volatile, printed listings of its current values on session and update reports replace formal backup copies.

The frequency with which backup copies of files should be made for files that require them cannot be estimated accurately during system design, as it depends on how SPECS is used. In general, the files should be backed up whenever the user prefers paying for making the backup copy to redoing all changes made since the last backup copy was made.

9.2.1 Old versions of the master file

During each batch update, the current version of the master file is superceded by a new version. The two most recently superceded versions are kept. Either of these versions--the "old master file" that was updated to produce the current master file and the "old, old master file" updated to produce the old master file--can be used to reconstruct the current master file.

9.2.2 Transaction-file backups

SPECS keeps current transaction-file backups to reconstruct the current transaction file, and old and cumulative backups to reconstruct the current master file.

Current transaction-file backups are simply copies of the current transaction file, made at times chosen by the users. One copy of the current backup file is needed, and may be replaced by a copy of a later version of the same file. Current transaction-file backups may be discarded when the transaction file is used in a batch update.

Old transaction-file backups are the transaction files used to update the old and old, old master files. These backup files must be retained while the contemporaneous master files are kept.

Cumulative transaction-file backups include all of the transactions used in batch updates. To maintain them, SPECS appends the transaction file from each update to the end of the current cumulative file. Cumulative transaction files are retained in duplicate for the lifetime of the SPECS system.

9.2.3 Site-yr-acc file backups

Backup files for the site-yr-acc file are simply copies of the contents of the file at various times. Each copy is kept until it is superceded by a newer one.

The effort that is justified for making backup copies of the site-yr-acc file depends on the RLA's data-entry strategy. If the RLA builds the site-yr-acc file for report-generating or uses SPECS for specimens from surveys, the site-yr-acc file may have so many entries that re-entering them is more effort than making backup files. If, by contrast, the RLA enters only excavation data from a few sites and years, making backup copies may take more effort than re-entering all of the information in the files; in this case, backup files should not be made.

9.2.4 Program-file backups

The backup files needed for SPECS programs depend on the SPECS implementation, and cannot be discussed in detail here. At a minimum, machine-readable copies and listings of all current source code must be kept. The implementor may suggest other backup files.

9.3 RECONSTRUCTING FILES

Files that are lost or damaged can be reconstructed, often from the backup files and outdated versions described in the last section of this chapter. This section describes how SPECS reconstructs lost files.

9.3.1 Reconstructing the master file

Master files are reconstructed by repeating one or more of the batch updates that contributed to the lost file. To recreate a lost file, the user must reset the "file ID" and "next transaction number" in the progress record of the system parameters file before and after master-file reconstruction. These values can be reset during interactive sessions.

9.3.1.1 Reconstruction from outdated files

If the old master file and the contemporaneous transaction file are available, the current master file can be reconstructed by simply repeating the most recent batch update. The new transaction file created by the update should be discarded, as it duplicates the beginning of the current transaction file.

An alternative approach to master-file reconstruction from an old master file and the contemporaneous transaction file is to append the current transaction file to the old one, then use the resulting multiple-update transaction file to update the old master file.

The master file can be reconstructed from the old, old master file, the transaction files contemporaneous with it and with the old master file, and the current transaction file. In this reconstruction, the three transaction files are concatenated to form a multiple-update file for updating the old, old master file.

9.3.1.2 Reconstruction from the cumulative transaction file

Reconstructing the master file from the cumulative transaction file uses an extreme version of the techniques for reconstruction with multiple-update transaction files that were discussed in the last section. In this version, the master file to be updated has no specimen-data records and the multiple-update file is the combination of the cumulative and current transaction files.

9.3.1.3 Cautions

Master files created by the techniques of this chapter may not precisely duplicate the lost files, for five reasons:

1. In the techniques that use the current transaction file as part of a multiple-update file, the new master file includes the effects of transactions that were entered after the lost file was created.
2. If the validation constants have been reset since the oldest transactions used in the update were requested, batch-update validation may fail for different transactions during the reconstruction than it did when the lost file was created.
3. If the descriptions of specimens have changed since the oldest transactions used in the update were requested, change-data and delete-data records that have substrings to be matched in the descriptions of specimens they affect may be applied to different

specimen-data records during the reconstruction than they were during the creation of the lost file.

4. Insert-data transactions that failed validation for the lost file may pass it during reconstruction if different transactions are applied to them, or fail during reconstruction and have passed when the lost file was built. Thus, the reconstructed master-file may have different records than the lost file. The users must be careful to check for records that have been duplicated or inadvertently deleted in the reconstructed file.
5. Master-file key changes still require two updates to complete. The update used for reconstruction is the first, but a second may be needed to finish changes that were already completed in the lost file.

Because the lost and reconstructed files may be different, it is vital that the users read the update report for the reconstruction with great care.

9.3.2 Reconstructing the current transaction file

The current transaction file is reconstructed by copying the most recent current-transaction backup file, then re-entering any transactions that were written onto the lost transaction file after the backup copy was made. The transactions to be re-entered are shown in the reports from the interactive sessions that followed making the backup copy.

9.3.3 Reconstructing the site-yr-acc file

The technique for reconstructing the site-yr-acc file is similar to that for the current transaction file: copy the most recent backup file, then re-enter changes made after the backup file was created. If there is no backup file, all of the information on the lost file must be re-entered.

9.3.4 Reconstructing the system parameters file

Reconstructing the system parameters file is done by re-entering all of the values in it when SPECS requests them during the first interactive session after the file is lost. Batch updates cannot be run until the system parameters file is reconstructed, but data retrieval from the master, transaction, and site-yr-acc files is unaffected by the loss of the system parameters file.

The current values for the validation constants and progress information are printed on each session and update report. Current values of passwords can be gotten from the RLA.

9.3.5 Reconstructing the SPECS programs

The implementor is responsible for designing and documenting techniques for recreating SPECS from source code and any other backup files he specifies.

9.3.6 Recovering from the loss of backup files and old versions

9.3.6.1 The cumulative transaction file

If both copies of the cumulative transaction file are lost, they cannot be reconstructed. It is tempting to say that reconstruction can be done by re-entering the lost transactions; but it is likely that the re-entry would introduce new errors unless the cumulative transaction file was very small.

Recovering from the loss of both copies of the cumulative transaction file is probably best done by guaranteeing that the lost file will never be needed. This assurance is gained by permanently retaining duplicate copies of the current master file, then using the current transaction file as the beginning of a new cumulative file.

If only one copy of the cumulative transaction file is lost, a duplicate of the remaining copy should immediately be made; no further action is needed.

If all copies of the cumulative transaction file and all old and current versions of the master file are lost, no recovery is possible. This, arguably the greatest catastrophe that could befall SPECS, would mean that all of the data must be re-entered.

9.3.6.2 Other files

The loss of any backup file or old version other than the cumulative transaction file does not seriously affect SPECS.

Backup files should be replaced by making new copies of the files they are protecting. Old master and transaction files probably do not need replacement, unless there is reason to believe that all of the old versions and the cumulative transaction file are also endangered.

Chapter X

STATUS

This paper marks the end of system design for SPECS, which is now ready to implement. The appendices provide information to be used during implementation: specifications of input, command, and record formats and report entries, and pseudocode for interactive sessions and batch updates.

Appendix A

DATA AND COMMAND FORMATS

A.1 DATA FORMATS

SPECS uses two kinds of formats for specimen-catalog data: input formats and internal formats. The user enters the data in an input format, and SPECS stores it in an internal format. Data is converted from an input format to an internal format by single-field validation during interactive sessions. Batch updates and report generation use only data represented in an internal format.

In both input and internal formats, the data is divided into "fields," each containing a single kind of information about a single specimen. The information in each field is referred to as the "value" of the field. In the internal format, the fields are the fields in the records of the master, transaction, system parameters, and site-yr-acc files. These fields are enumerated in Appendix B.

The fields in the input format are the "input fields" that are briefly described in the rest of this section. The description of each field lists the information that is expected in the field, and perhaps the source of the information in the columns of the Specimen Catalog form (Figure 1). Detailed information about the input-field values is in Appendix C, which summarizes SPECS' assumptions about how the value of each input field is represented and converted to internal fields. Appendix E specifies how the values of the internal fields are combined to form the records that are written onto the SPECS files.

There are eleven input fields: "accession number," "count," "depth," "description," "grid," "location," "site," "specimen number," "unit," "weight," and "year."

An "accession number" field has an isolated accession number as its value. The user enters isolated accession numbers in SET-ACC commands and in response to prompts for accession numbers. Accession numbers that are considered as parts of specimen numbers are entered in the "specimen number" input field.

The value of a "count" input field specifies the number of things in a specimen and the kind of things that were counted (pieces or bags, for example). This information is usually found in the "Number" column of a Specimen Catalog form but, for some subspecimens, may be in the "Description" column instead. The value entered in a "count" field depends to some extent on how subspecimens are treated, a subject that is discussed in Section A.2.

The value of a "depth" field is the depth at which a specimen was collected. This information is found in the "Location" column of a Specimen Catalog form. SPECS recognizes three kinds of depths: surface, levels, and zones. Levels within zones are recorded in the "depth" field simply as zones; their subdivisions into levels are entered as part of the "location" input field.

The "description" input field is a catch-all for information describing the specimen. It certainly contains the specimen's description (found in the "Description" column of a Specimen Catalog form), but also includes any information about the specimen's count or weight that cannot be represented in the "count" and "weight" input fields. The "description" field may include information about subdivisions of the specimen, depending on how the RLA chooses to enter subspecimens.

A "grid square" input field contains information in the site's coordinate system. Grid-square information is found in the "Location" column of a Specimen Catalog form. The information may represent a point provenience, a grid square, or another kind of area.

The "location" input field is a catch-all for information from the "Location" column of the Specimen Catalog form that cannot be entered in the "depth," "grid," "site," and "unit" fields. Examples of such information are: levels within zones, depth measurements (as opposed to level and zone numbers, which are in the "depth" field), and descriptions of the soil surrounding a specimen. If SPECS finds that some of the data entered in the "depth," "grid," and "unit" fields cannot be converted into an internal format without losing information, it may treat the data as though it were entered in the "location" field.

The "site" field contains the number of the site at which the specimen was collected. For excavated sites, this information is usually in the "Site number" space at the top of each Specimen Catalog form; for survey sites, it is often in the "Location" column.

The value of the "specimen number" field is a specimen number. The accession number is often entered as part of the specimen number, but may sometimes be omitted when the INPUT command is used. The specimen type code, sequence number, and continuation number (if there is one) are enterd from the "Spec. No." column of the Specimen Catalog form.

The "unit" input field contains information about the burial, feature, structure, post hole, test pit, or other kind of location that is not defined by the site's coordinate system. This usage of "unit" differs from that of Chapter 2 in that the locations identified in the "unit" field are often subdivided along level, zone, or grid-square boundaries. The information in the "unit" input field is from the "Location" column of the Specimen Catalog form.

The "weight" input field contains the weight of the specimen, in ounces. This information is found in either the "Description" or the "Count" column of the Specimen Catalog Form.

The value of the "year" input field is the year during which the specimen was collected.

A.1.1 Entering data about new specimens

Data about new specimens is given to SPECS in the "new-specimen" input format, which allows the user to enter all of the information about a specimen at one time. The INPUT command tells SPECS to prepare for information in the new-specimen format.

The new-specimen input format has eight columns, containing values for the specimen number, unit, grid, location, count, weight, and description input fields. When SPECS is given an INPUT command, it prints headers identifying the columns. The column headers are Terminal Report 4, in Appendix D.

Each of the columns must have a value for each specimen. If the specimen has no information for a column, the value of the column is entered as NONE.

When the user enters data about a new specimen, he must separate the values for different input fields by two or more blanks. SPECS uses these blanks, rather than the positions in which the values are typed at the terminal, to recognize the fields to which the values belong. The column headers remind the user of the order in which the values should be given, but do not dictate to him where they must be typed.

If SPECS finds what appear to be values for eight input fields, it assumes that the fields are in the order given by the column headers. Otherwise, it asks the user to identify the input field to which each of the apparent values belongs, then prompts him for the value of any fields whose values are still unknown.

The user can continue input fields whose values do not fit in the space available for them on a single line by breaking them between lines at blanks in their values. SPECS assumes that any field whose value ends with a "+" is continued on the next line.

The conventions by which SPECS associates continuations with fields to be continued are similar to those for values and input fields: if it finds the number of continuations it expects, it assumes that they are in the same order as the fields needing continuations; and if it does not, it asks the user for help. When SPECS joins a continuation with the value being continued, it removes the "+" and leaves a single blank between the currently known value of the field and the continuation.

SPECS treats information about the first specimen entered after an INPUT command differently from information about later specimens. This difference is apparent for information in the "specimen number," "site," and "year" input fields, and in the use of ditto marks. The validation for the specimen is also slightly different: no sequence checking is performed during interactive sessions for the first specimens after INPUT commands.

The user types a specimen number in the "specimen number" input field. He may include an accession number whenever he wishes, but is required to do so only for the first specimen after an INPUT command. If he does not give an accession number for later specimens, SPECS assumes that their accession numbers are the same as the last accession number it was told. When an accession number is needed but not given, SPECS asks the user for it.

Values for the "site" and "year" input fields are not entered in the columns of the new-specimen input format. When SPECS needs information about sites and years, it asks the user for it. SPECS asks for this information for the first specimen entered after an INPUT command and for any specimen for which the user has given an accession number. Sites and years, like accession numbers, are assumed to have the last values that were given for them. A user who wants to change the value of the "site" or "year" field should induce SPECS to prompt him for their values by entering an accession number.

Ditto marks may be used to enter data about any specimen that is not the first after an INPUT command. When SPECS finds an input field whose value is a single ditto mark ("), it assumes the value of the input field is the same as it was for the last specimen to be entered. If ditto marks are used for the first specimens after INPUT commands, SPECS insists that the user tell it the values of the fields for which the ditto marks were used.

A.2 SPECIMEN DEFINITION

The specimens entered in SPECS may differ from those in the specimen catalog in two cases: subspecimens and commands.

A.2.1 Subspecimens

Subspecimens, presented in Chapter 2 as subdivisions of specimens for which separate counts and descriptions were recorded, may be treated in SPECS as distinct specimens having the same specimen number. Alternatively, the distinction between the subspecimens of a single specimen may be considered part of the specimen's description. The choice between these treatments influences the ease with which the users can retrieve various kinds of information from SPECS.

Information about subspecimens represented as separate specimens is retrieved like information about any other specimens. Data about the entire specimen is harder to retrieve, however: total counts and weights can be calculated from the subspecimen counts and weights (though it is a nuisance to do so), but deriving an overall description of the specimen from subspecimen descriptions is far more awkward.

Another disadvantage of representing subspecimens separately occurs because the subspecimens have the same location information. When the user changes location information for one subspecimen, he must be careful to make the same change for all of the others. As described in Chapter 5, the user may specify in a CHANGE command that all subspecimens of a single specimen should be changed at once.

Both of these disadvantages of separately represented specimens are avoided by relegating the distinction between subspecimens to the description of the specimen. The total counts and weights and the overall description need not be derived because they are recorded as the specimen's count, weight, and description. Because the location information is recorded only once, it cannot be made inconsistent by changes that affect only some subspecimens.

But representing subspecimens in the specimen description has the major disadvantage that information about subspecimens is very awkward to retrieve. If the user knows precisely how each subspecimen in which he is interested is represented in the SPECS files, he may be able to retrieve the information he wants. If, as is more likely, he does not know their representations, the user may have to re-enter the subspecimens separately before he can retrieve them.

The RLA must develop a policy for subspecimen representation. The policy may tailor the representation to the kind of specimen being represented. For example, the RLA may consider it more useful to know that a specimen has 25 large potsherds and 30 small ones than to know that it has 55 sherds. At the same time, it may want to know that a specimen has 23 beads, but not that it has 5 white beads, 6 blue beads, and 12 striped red beads. Given these preferences, the RLA should represent subspecimens of potsherds as separate specimens and subspecimens of beads as details in the description of a single specimen. Whatever its decision, the RLA must be consistent, since data entered with a single technique is apt to be easier to retrieve than data entered with a hodgepodge of techniques.

A.2.2 Comments

The RLA enters comments about locations in the specimen catalog without assigning them specimen numbers. SPECS considers these comments as specimens of the new type "c" (for "comment").

When a user enters a comment, he provides it with a specimen number that preserves its position in the specimen catalog. For example, a comment listed between specimens 1002a13/2 and 1002a14 might be numbered 1002c13/3.

A.3 COMMAND FORMATS

SPECS recognizes the twelve commands whose formats are specified in this section. The meanings of the commands are summarized here, and are described in Chapter 5.

Four notational conventions used in this section are:

1. Anything that is enclosed by square brackets may be omitted at the user's discretion. The meanings of the bracketed parts of each command are described when the command format is listed.
2. Punctuation marks and capital letters are as they should be typed by the user.
3. Anything that is underlined will be replaced by a value appropriate for the command. The meanings of the underlined parts of each command are described when the command format is listed.
4. The user may truncate the first words of the commands, but must give enough of the words to make the commands uniquely recognizable. For example, the CHANGE command can be entered as CHA, CHAN, CHANG, or CHANGE; but it cannot be given as CH (which could also be a short form of CHECK).

If the user enters only part of the information shown in the command formats, SPECS prompts him for any information it needs. The user must enter only the first word of each command to have the command executed.

A.3.1 The CANCEL command

CANCEL trans# FOR spec#

CANCEL cancels an earlier command by telling SPECS to ignore a transaction-file record written by the command to be cancelled.

In the CANCEL command,

trans# is the transaction number of the transaction to be ignored. This number is given in the session-report entry for the command to be cancelled.

spec# is the specimen number, including accession number, of the specimen for which information would be affected if the transaction written by the cancelled command were not ignored.

A.3.2 The CHANGE command

CHANGE [ALL] field FOR spec# [WITH "description"]

The CHANGE command changes specimen-catalog data given by an earlier command, whether or not the data has already become part of the master file. When SPECS receives a CHANGE command, it asks the user for new information to replace the data to be changed.

In this command,

<u>field</u> is the name of the input field in which the value to be changed was entered.

<u>spec#</u> is the number of the specimen for which the data to be changed was given.

<u>description</u> is a part of the description of the specimen, as presented in Chapter 5.

[ALL] and [WITH "<u>description</u>"] are used as discussed in Chapter 5. [ALL] has no effect in the same CHANGE command as [WITH "<u>description</u>"].

A.3.3 <u>The CHECK command</u>

CHECK <u>trans#</u> FOR <u>spec#</u>

The CHECK command tells SPECS to not to execute a transaction that changes specimen-catalog data unless the changed information passes batch-update validation.

In this command,

<u>trans#</u> is the transaction number of the transaction that changes the information. This number is given on the session report for the session by which the transaction was written.

<u>spec#</u> is the specimen number, including accession number, of the specimen for which transaction <u>trans#</u> changes information.

A.3.4 The COMMENT command

COMMENT comment

The COMMENT command tells SPECS to write a message given by the user onto the session report.

In this command,

comment is the message to be written on the session report.

A.3.5 The CONFIRM command

CONFIRM trans# FOR spec#

The CONFIRM command confirms that SPECS should execute an earlier transaction that changes specimen-catalog data, even if the changed information fails batch-update validation.

In this command,

trans# is the transaction number of the transaction that changes the information. This number is given in the session report for the session by which the transaction was written.

spec# is the specimen number, including accession number, of the specimen for which transaction trans# changes information.

A.3.6 <u>The DELETE command</u>

DELETE [ALL] <u>spec#</u> [WITH "<u>description</u>"]

The DELETE command deletes all of the information about one or more specimens with a common specimen number, whether or not the information has already become part of the master file.

In this command,

<u>spec#</u> is the specimen number for which information is to be deleted.

<u>description</u> is a part of the description of the specimen, as presented in Chapter 5.

[ALL] and [WITH "<u>description</u>"] are used as discussed in Chapter 5. [ALL] has no effect in the same DELETE command as [WITH "<u>description</u>"].

A.3.7 <u>The END command</u>

END

The END command tells SPECS to end the session because the user has finished giving commands.

A.3.8 <u>The INPUT command</u>

INPUT

The INPUT command tells SPECS to prepare for specimen-catalog data about one or more new specimens. The data is not part of the command, but is entered in the new-specimen format described in section 1.1 of this appendix.

A.3.9 The RESET command

RESET name

The RESET command resets a password or validation constant whose value is stored in the system parameters file. When SPECS receives a RESET command, it asks the user for a new value for the password or constant.

In this command,

name is the name of the password or validation constant to be reset. The passwords and validation constants are listed in Appendix B. SPECS also recognizes short forms of the password and constant names, as listed in the menu printed for step 5.14.1 in the interactive pseudocode (pages 482 to 486).

A.3.10 The SET-ACC command

SET-ACC [FOR] site(year) [TO] acc#

The SET-ACC command enters or changes an entry in the site-yr-acc file for a single site, year, and accession number, or deletes all entries for a site and year.

In this command,

- site is a site number.
- year is a year.
- acc# is an accession number to be entered for the site and year, or NONE if the site and year are to be deleted.
- [FOR] and [TO] may be typed to help the user remember the order in which the site, year, and accession number must be given. They have no effect on the execution of the command.

A.3.11 The SHOW-ACC command

SHOW-ACC [FOR] site(year)

The SHOW-ACC command prints any accession numbers in the site-yr-acc file for a site and year. If there are none, it prints a message to tell the user so.

In this command,

- site is a site number.
- year is a year.
- [FOR] may be typed to parallel the syntax of the SET-ACC command, but has no effect on the execution of the SHOW-ACC command.

Appendix B

RECORDS AND FIELDS

This appendix enumerates the formats and contents of records in the files retained by SPECS. It does not include report files, which are discussed in Chapter 4 and Appendix D.

The appendix has four sections, each containing information about a single file. Each section is further divided into subsections describing single kinds of records. The information given about each record includes the names, sizes, formats, and values of each of its fields. Table 3 shows the relationship between files, records, and subsections of the appendix.

SPECS represents numbers as two's-complement integers, each requiring one to three bytes. In this representation, a 1-byte number may range from -128 to 127, a 2-byte number may range from -32,768 to 32,767, and a 3-byte number may range from -8,338,608 to 8,338,607. Fractions are multiplied by the appropriate power of ten to convert them to integers.

Characters are each represented in one byte.

These representations of numbers and characters are convenient in PL/I (as FIXED BINARY and CHARACTER), and in SAS (as IB and $) [references 2: 17, 21; 5: 17; 7: 34, 36].

TABLE 3

Contents of Appendix B

File name	Record name	Subsection
Master	Master-file header	B.1.1
	Specimen-data	B.1.2
Transaction	Transaction-file header	B.2.1
	Insert-data	B.2.2
	Change-data	B.2.3
	Delete-data	B.2.4
	Cancel-transaction	B.2.5
	Validation-override	B.2.6
	Validation-restoration	B.2.7
Site-yr-acc	Site-yr-acc	B.3.1
System parameters	Progress	B.4.1
	Validation constants	B.4.2
	Passwords	B.4.3

B.1 RECORDS IN THE MASTER FILE

B.1.1 The Master-file Header

Record size: 15 bytes.

Fields:

Accession number.

Format: Numeric.

Field size: 2 bytes.

Value: 0000.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

Format: Numeric.

Field size: 3 bytes.

Value: 000000.

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

Format: Character.

Field size: 1 byte.

Value: M.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Master-file ID.

Format: Numeric.

Field size: 3 bytes.

Value: the current transaction number when the file was created; also, the ID of the transaction file to be used to update this master file.

Old-master-file ID.

<u>Format</u>: Numeric.

<u>Field size</u>: 3 bytes.

<u>Value</u>: the master-file ID of the master file that was updated to create this master file; also, the ID of the transaction file used in the update.

File creation date.

<u>Format</u>: Numeric.

<u>Field size</u>: 3 bytes.

<u>Value</u>: the date of the batch update creating this file, represented as YYMMDD. In this representation, the first two digits are the year, the next two are the month (01 to 12), and the last two are the day of the month (01 to 31).

B.1.2 The Specimen-data Record

<u>Record size</u>: variable; 24 to 292 bytes.

<u>Fields</u>:

Accession number.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: an accession number.

This field is part of the master-file key.

Sequence and continuation number.

<u>Format</u>: Numeric.

<u>Field size</u>: 3 bytes.

<u>Value</u>: a sequence number, including continuation number.

This field is part of the master-file key.

Specimen type.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Value</u>: the specimen type code (a, b, c, e, h, m, or p) from the specimen number.

This field is part of the master-file key.

Site.

<u>Format</u>: Character.

<u>Field size</u>: 9 bytes.

<u>Value</u>: the site from which the specimen was collected.

Year.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: the year during which the specimen was collected.

Unit type.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Values</u>:

F for features

B for burials

P for post holes

T for test pits

S for structures

N if there is no unit

O for other

This field indicates the type of unit information recorded.

Grid-square type.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Values</u>:

G for normal grid squares

P for point proveniences

A for areas denoted by ranges of coordinates

N if there is no grid square

O for other

This field indicates the type of grid-square information recorded.

Depth type.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Values</u>:

L for levels

Z for zones

S for surface

N if there is no depth information

O for other

This field indicates the kind of depth information recorded.

Count type.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Values</u>:

B for bags

J for jars

X for boxes

V for vials

M for mixed items

P for pieces

N if there is no count

O for other

This field indicates the kind of items that were counted.

Weight type.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Values</u>:

G if the true weight is greater than the recorded weight

L if the true weight is less than the recorded weight

E if the true weight is the recorded weight

N if there is no recorded weight

This field indicates whether the correct weight was measured and entered, and if not, the direction of the error.

Location length.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: the length in bytes of the "location" field; also, the number of characters in the location information.

Description length.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: the length in bytes of the "description" field; also, the number of characters in the description information.

Unit.

If the unit type is N or O, there is no "unit" field.

Otherwise, the "unit" field has these characteristics:

Format: Numeric.

Field size: 2 bytes.

Value: the numeric part of the unit number.

N-S coord.

If the grid-square type is N, there is no "N-S coord" field.

Otherwise, the "N-S coord" field has these characteristics:

Format: Numeric.

Field size: 2 bytes.

Value: the north-south coordinate of a point.

L-R half.

If the grid-square type is N, there is no "L-R half" field.

Otherwise, the "L-R half" field has these characteristics:

Format: Character.

Field size: 1 byte.

Value: the "L" or "R" that signifies the left or right half of the site's coordinate system.

E-W coord.

If the grid-square type is N, there is no "E-W coord" field.

Otherwise, the "E-W coord" field has these characteristics:

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: the east-west coordinate of a point.

Depth.

If the depth type is N, O, or S, there is no "depth" field.

Otherwise, the "depth" field has these characteristics:

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 3 bytes.

<u>Value</u>: A level or zone number.

Count.

If count type is N, there is no "count" field.

Otherwise, the "count" field has these characteristics:

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The number of items, of the kind indicated by the count type, in the specimen.

Weight.

If the weight type is N, there is no "weight" field. Otherwise, the "weight" field has these characteristics:

Format: Numeric.

Field size: 2 bytes.

Value: The weight of the specimen, in tenths of ounces.

Location.

Format: Character.

Field size: The value of "location length," interpreted as a size in bytes.

Value: any location information that, because of its format or content, does not fit into another field.

Description.

Format: Character.

Field size: the value of "description length," interpreted as a size in bytes.

Value: a description of the specimen being recorded.

B.2 RECORDS IN THE TRANSACTION FILE

B.2.1 The transaction-file header

Record size: 12 bytes.

Fields:

Accession number.

Format: Numeric.

Field size: 2 bytes.

Value: 0000.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

Format: Numeric.

Field size: 3 bytes.

Value: 000000

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

Format: Character.

Field size: 1 byte.

Value: T.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Xaction-file ID.

Format: Numeric.

Field size: 3 bytes.

Value: the current transaction number when the file was created; also, the ID of the master file to be updated by this transaction file.

File creation date.

Format: Numeric.

Field size: 3 bytes.

Value: the date of the batch update creating this file, represented as YYMMDD. In this representation, the first two digits are the year, the next two are the month (01 to 12), and the last two are the day of the month (01 to 31).

B.2.2 *The insert-data record*

Record size: variable; 28 to 296 bytes.

Fields:

Accession number.

Format: Numeric.

Field size: 2 bytes.

Value: an accession number.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

<u>Format</u>: Numeric.

<u>Field size</u>: 3 bytes.

<u>Value</u>: a sequence number, including continuation number.

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Value</u>: the specimen type code (a, b, c, e, h, m, or p) from the specimen number.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Transaction number.

<u>Format</u>: Character.

<u>Field size</u>: 3 bytes.

<u>Value</u>: the unique transaction number assigned by SPECS when this record was written.

Transaction type code.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: I.

This field indicates that this is an insert-data record.

Site.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 9 bytes.

<u>Value</u>: the site from which the specimen was collected.

Year.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: the year during which the specimen was collected.

Unit type.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Values</u>:

F for features

B for burials

P for post holes

T for test pits

S for structures

N if there is no unit

O for other

This field indicates the type of unit information recorded.

Grid-square type.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Values</u>:

G for normal grid squares

P for point proveniences

A for areas denoted by ranges of coordinates

N if there is no grid square

O for other

This field indicates the type of grid-square information recorded.

Depth type.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Values</u>:

L for levels

Z for zones

S for surface

N if there is no depth information

O for other

This field indicates the kind of depth information recorded.

Count type.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Values</u>:

B for bags

J for jars

X for boxes

V for vials

M for mixed items

P for pieces

N if there is no count

O for other

This field indicates the kind of items that were counted.

Weight type.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Values</u>:

G if the true weight is greater than the recorded weight

L if the true weight is less than the recorded weight

E if the true weight is the recorded weight

N if there is no recorded weight

This field indicates whether the correct weight was measured and entered, and if not, the direction of the error.

Location length.

<u>Format</u>: Numeric.

<u>Field size</u>: 1 byte.

<u>Value</u>: the length in bytes of the "location" field; also, the number of characters in the location information.

Description length.

<u>Format</u>: Numeric.

<u>Field size</u>: 1 byte.

<u>Value</u>: the length in bytes of the "description" field; also, the number of characters in the description information.

Unit.

If the unit type is N or O, there is no "unit" field.

Otherwise, the "unit" field has these characteristics:

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: the numeric part of the unit number.

N-S coord.

If the grid-square type is N, there is no "N-S coord" field.

Otherwise, the "N-S coord" field has these characteristics:

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: the north-south coordinate of a point.

L-R half.

If the grid-square type is N, there is no "L-R half" field.

Otherwise, the "L-R half" field has these characteristics:

Format: Character.

Field size: 1 byte.

Value: the "L" or "R" that signifies the left or right half of the site's coordinate system.

E-W coord.

If the grid-square type is N, there is no "E-W coord" field.

Otherwise, the "E-W coord" field has these characteristics:

Format: Numeric.

Field size: 2 bytes.

Value: the east-west coordinate of a point.

Depth.

If the depth type is N, O, or S, there is no "depth" field.

Otherwise, the "depth" field has these characteristics:

Format: Character.

Field size: 3 bytes.

Value: A level or zone number.

Count.

If the count type is N or O, there is no "count" field.

Otherwise, the "count" field has these characteristics:

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: The number of items, of the type indicated by the count type, in the specimen.

Weight.

If the weight type is N, there is no "weight" field.

Otherwise, the "weight" field has these characteristics:

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: The weight of the specimen, in tenths of ounces.

Location.

<u>Format</u>: Character.

<u>Field size</u>: The value of "location length," interpreted as a size in bytes.

<u>Value</u>: any location information that, because of its format or content, does not fit into another field.

Description.

Format: Character.

Field size: the value of "description length," interpreted as a size in bytes.

Value: a description of the specimen being recorded.

B.2.3 The change-data record

Record size: variable; 16 to 288 bytes.

Fields:

Accession number.

Format: Numeric.

Field size: 2 bytes.

Value: an accession number.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

Format: Numeric.

Field size: 3 bytes.

Value: a sequence number, including continuation number.

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: the specimen type code (a, b, c, e, h, m, or p) from the specimen number.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Transaction number.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 3 bytes.

<u>Value</u>: the unique transaction number assigned by SPECS when this record was written.

Transaction type code.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: C.

This field indicates that this is a change-data record.

Transaction options.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>:

-1 if the user gave no information about which specimen-data records with duplicate master-file keys should be affected.

0 if the user said that all such records should be changed.

1 to 127 if the user gave a substring to be matched in the "description" fields of the records to be affected. In this case, the value of "transaction options" is the number of characters in the substring.

Input field.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: a code from Table 4.

This field indicates from what input field the changed values come.

Field count.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: the number of specimen-data fields to be changed in each record affected by this transaction.

TABLE 4

Input fields and codes for change-data records

Code	Field
1	Count
2	Depth
3	Description
4	Grid square
5	Location
6	Site
7	Specimen number
8	Unit
9	Weight
10	Year

Description.

If transaction options is < 1, there is no "description" field.

Otherwise, the "description" field has these characteristics:

Format: Character.

Field size: The value of "transaction options," interpreted as a size in bytes.

Value: a substring to be matched in the "description" fields of all specimen-data records affected by this transaction.

A new-field description for each field to be changed in the specimen-data record. A new-field description is composed of a "field designation," a "field size," a "field format," and perhaps a "new value" field for each field to be changed. The number of new-field descriptions in each change-data record is the value of the "field count" field in the record.

Field designation.

<u>Format</u>: Numeric.

<u>Field size</u>: 1 byte.

<u>Value</u>: a code designating the specimen-data field to be changed, from Table 5.

TABLE 5

Field-designation codes for change-data records

Code	Field
01	Accession number
02	Sequence and continuation number
03	Specimen type
04	Site
05	Year
06	Unit type
07	Grid-square type
08	Depth type
09	Count type
10	Weight type
11	Unit

12	N-S coord
13	L-R half
14	E-W coord
15	Depth
16	Count
17	Weight
18	Location (replace value)
19	Description (replace value)
20	Location (concatenate value)
21	Description (concatenate value)

Field size.

Format: Numeric.

Field size: 1 byte.

Value: The size, in bytes, of the field whose value is being changed, as given in the description of the specimen-data record.

Field format.

Format: Character.

Field size: 1 byte.

Value:

"N" if the format of the field whose value is being changed is given in the description of the specimen-data record as numeric.

"C" if the format of the changed field is character.

New value.

If "field size" is < 1, there is no "new value" field.

Otherwise, the "new value" field has these characteristics:

Format: as specified by the value of "field format".

Field size: as specified by the value of "field size".

Value: a new value for the field indicated by the value of "field designation".

B.2.4 The delete-data record

Record size: variable; 11 to 138 bytes.

Fields:

Accession number.

Format: Numeric.

Field size: 2 bytes.

Value: an accession number.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

Format: Numeric.

Field size: 3 bytes.

Value: a sequence number, including continuation number.

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

Format: Character.

Field size: 1 byte.

Value: the specimen type code (a, b, c, e, h, m, or p) from the specimen number.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Transaction number.

Format: Numeric.

Field size: 3 bytes.

Value: the unique transaction number assigned by SPECS when this record was written.

Transaction type code.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Value</u>: D.

This field indicates that this is a delete-data record.

Transaction options.

<u>Format</u>: Numeric.

<u>Field size</u>: 1 byte.

<u>Value</u>:

-1 if the user gave no information about which specimen-data records with duplicate master-file keys should be affected.

0 if the user said that all such records should be deleted.

1 to 127 if the user gave a substring to be matched in the "description" fields of the records to be affected. In this case, the value of "transaction options" is the number of characters in the substring.

Description.

If transaction options is < 1, there is no "description" field.

Otherwise, the "description" field has these characteristics:

<u>Format</u>: Character.

<u>Field size</u>: The value of "transaction options," interpreted as a size in bytes.

<u>Value</u>: a substring to be matched in the "description" fields of all specimen-data records to be affected by this transaction.

B.2.5 The cancel-transaction record

<u>Record size</u>: 13 bytes.

<u>Fields</u>:

Accession number.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: an accession number.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

Format: Numeric.

Field size: 3 bytes.

Value: a sequence number, including continuation number.

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

Format: Character.

Field size: 1 byte.

Value: the specimen type code (a, b, c, e, h, m, or p) from the specimen number.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Transaction number.

Format: Numeric.

Field size: 3 bytes.

Value: the unique transaction number assigned by SPECS when this record was written.

Transaction type code.

Format: Character.

Field size: 1 byte.

Value: X.

This field indicates that this is a cancel-transaction record.

Affected transaction number.

Format: Numeric.

Field size: 3 bytes.

Value: The number of the transaction to be cancelled.

B.2.6 The validation-override record

Record size: 13 bytes.

Fields:

Accession number.

Format: Numeric.

Field size: 2 bytes.

Value: an accession number.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

Format: Numeric.

Field size: 3 bytes.

Value: a sequence number, including continuation number.

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

Format: Character.

Field size: 1 byte.

Value: the specimen type code (a, b, c, e, h, m, or p) from the specimen number.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Transaction number.

Format: Numeric.

Field size: 3 bytes.

Value: the unique transaction number assigned by SPECS when this record was written.

Transaction type code.

<u>Format</u>: Character.

<u>Field size</u>: 1 byte.

<u>Value</u>: 0.

This field indicates that this is a validation-override record.

Affected transaction number.

<u>Format</u>: Numeric.

<u>Field size</u>: 3 bytes.

<u>Value</u>: The number of the transaction for which validation is to be overridden.

B.2.7 <u>The validation-restoration record</u>

<u>Record size</u>: 13 bytes.

<u>Fields</u>:

Accession number.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: an accession number.

As part of the master-file key, this field corresponds to the "accession number" field in the specimen-data record.

Sequence and continuation number.

Format: Numeric.

Field size: 3 bytes.

Value: a sequence number, including continuation number.

As part of the master-file key, this field corresponds to the "sequence and continuation number" field in the specimen-data record.

Specimen type.

Format: Character.

Field size: 1 byte.

Value: the specimen type code (a, b, c, e, h, m, or p) from the specimen number.

As part of the master-file key, this field corresponds to the "specimen type" field in the specimen-data record.

Transaction number.

Format: Numeric.

Field size: 3 bytes.

Value: the unique transaction number assigned by SPECS when this record was written.

Transaction type code.

Format: Character.

Field size: 1 byte.

Value: R.

This field indicates that this is a validation-restoration record.

Affected transaction number.

Format: Numeric.

Field size: 3 bytes.

Value: The number of the transaction for which validation is to be restored.

B.3 RECORDS IN THE SITE-YR-ACC FILE

B.3.1 The site-yr-acc record

Record size: 13 bytes.

Fields:

Site.

Format: Character.

Field size: 9 bytes.

Value: a site number.

This field is part of the key for the site-yr-acc file.

Year.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: a year.

This field is part of the key for the site-yr-acc file.

Accession number.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: an accession number assigned to samples from the site and year represented by the values of "site" and "year".

B.4 RECORDS IN THE SYSTEM PARAMETERS FILE

B.4.1 The progress record

<u>Record</u> <u>size</u>: 7 bytes.

<u>Fields</u>:

Record type.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 1 byte.

<u>Value</u>: 1

This field is the key for the system parameters file.

File ID.

<u>Format</u>: Numeric.

<u>Field size</u>: 3 bytes.

<u>Value</u>: The file identification of the master and transaction files to be used in the next batch update. This field corresponds to the "master-file ID" and "transaction-file ID" fields of the master-file and transaction-file header records.

Next transaction.

<u>Format</u>: Numeric.

<u>Field size</u>: 3 bytes.

<u>Value</u>: The next transaction number to be assigned.

B.4.2 <u>The validation-constants record</u>

<u>Record size</u>: 69 bytes.

<u>Fields</u>:

Record type.

<u>Format</u>: Numeric.

<u>Field size</u>: 1 byte.

<u>Value</u>: 2

This field is the key for the system parameters file.

Soil-sample weight.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: the maximum correct weight (in tenths of ounces) for a soil sample.

Washings weight.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: the maximum correct weight (in tenths of ounces) for washings.

Bead count.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of beads in a single specimen.

Type-"a" count.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces other than beads in a single specimen of type "a".

Type-"b" count.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces in a single specimen of type "b".

Type-"e" count.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces in a single specimen of type "e".

Type-"h" count.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces in a single specimen of type "h".

Flake count.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of flakes and chips in a single specimen of type "m".

Type-"m" count.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces other than flakes and chips in a single specimen of type "m" other than soil samples and washings.

Type-"p" count.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces in a single specimen of type "p".

Container count.

<u>Format</u>: Numeric.

<u>Field size</u>: 1 byte.

<u>Value</u>: The maximum correct value of "count" for specimens counted in boxes, jars, bags, or vials.

Maximum bead density.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of beads per tenth of an ounce.

Maximum type-"a" density.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces other than beads per tenth of an ounce in specimens of type "a".

Maximum type-"b" density.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces per tenth of an ounce in specimens of type "b".

Maximum type-"e" density.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces per tenth of an ounce in specimens of type "e".

Maximum type-"h" density.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The maximum correct number of pieces per tenth of an ounce in specimens of type "h".

Maximum flake density.

Format: Numeric.

Field size: 2 bytes.

Value: The maximum correct number of flakes and chips per tenth of an ounce in specimens of type "m".

Maximum type-"m" density.

Format: Numeric.

Field size: 2 bytes.

Value: The maximum correct number of pieces other than flakes and chips per tenth of an ounce in specimens of type "m" other than soil samples and washings.

Maximum type-"p" density.

Format: Numeric.

Field size: 2 bytes.

Value: The maximum correct number of pieces per tenth of an ounce in specimens of type "p".

Minimum bead density.

Format: Numeric.

Field size: 2 bytes.

Value: The minimum correct number of beads per tenth of an ounce.

Minimum type-"a" density.

Format: Numeric.

Field size: 2 bytes.

Value: The minimum correct number of pieces other than beads per tenth of an ounce in specimens of type "a".

Minimum type-"b" density.

Format: Numeric.

Field size: 2 bytes.

Value: The minimum correct number of pieces per tenth of an ounce in specimens of type "b".

Minimum type-"e" density.

Format: Numeric.

Field size: 2 bytes.

Value: The minimum correct number of pieces per tenth of an ounce in specimens of type "e".

Minimum type-"h" density.

Format: Numeric.

Field size: 2 bytes.

Value: The minimum correct number of pieces per tenth of an ounce in specimens of type "h".

Minimum flake density.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The minimum correct number of flakes and chips per tenth of an ounce in specimens of type "m".

Minimum type-"m" density.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The minimum correct number of pieces other than flakes and chips per tenth of an ounce in specimens of type "m" other than soil samples and washings.

Minimum type-"p" density.

<u>Format</u>: Numeric.

<u>Field</u> <u>size</u>: 2 bytes.

<u>Value</u>: The minimum correct number of pieces per tenth of an ounce in specimens of type "p".

Maximum level.

<u>Format</u>: Character.

<u>Field size</u>: 3 bytes.

<u>Value</u>: The maximum correct level or zone number.

Maximum E-W for "L".

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: the maximum (farthest west) correct value for an east-west coordinate in the left half of a site.

Minimum E-W for "L".

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: the minimum (farthest east) correct value for an east-west coordinate in the left half of a site.

Maximum E-W for "R".

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: the maximum (farthest east) correct value for an east-west coordinate in the right half of a site.

Minimum E-W for "R".

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: the minimum (farthest west) correct value for an east-west coordinate in the right half of a site.

Maximum N-S.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: The maximum (farthest north) correct value for a north-south coordinate.

Minimum N-S.

<u>Format</u>: Numeric.

<u>Field size</u>: 2 bytes.

<u>Value</u>: The minimum (farthest south) correct value for a north-south coordinate.

B.4.3 The passwords record

<u>Record size</u>: 73 bytes.

<u>Fields</u>:

Record type.

<u>Format</u>: Numeric.

<u>Field size</u>: 1 byte.

<u>Value</u>: 3

This field is the key for the system parameters file.

Master password.

<u>Format</u>: Character.

<u>Field size</u>: 8 bytes.

<u>Value</u>: the correct master password.

Input password.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 8 bytes.

<u>Value</u>: the correct input password.

Change password.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 8 bytes.

<u>Value</u>: the correct change password.

Delete password.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 8 bytes.

<u>Value</u>: the correct delete password.

Cancel password.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 8 bytes.

<u>Value</u>: the correct cancel password.

Validation-control password.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 8 bytes.

<u>Value</u>: the correct validation-control password.

Show-acc password.

<u>Format</u>: Character.

<u>Field</u> <u>size</u>: 8 bytes.

<u>Value</u>: the correct show-acc password.

Set-acc password.

Format: Character.

Field size: 8 bytes.

Value: the correct set-acc password.

Update password.

Format: Character.

Field size: 8 bytes.

Value: the correct update password.

Appendix C

VALIDATION

This appendix includes all validation algorithms for specimen-catalog data. It has three sections: single-field validation, multiple-field validation, and sequence checking. Single-field validation is performed during interactive sessions only; multiple-field validation and sequence checking are performed during both interactive sessions and batch updates.

The sections in the appendix are divided into subsections, each of which describes a kind of validation and presents an algorithm for it. The descriptions of single-field validation also specify the input formats expected for the information to which they are applied.

The algorithms in this appendix may be viewed as subroutines for the interactive-session and batch-update pseudocode (Appendices E and F). They are specified in the notation used in the batch-update pseudocode, but produce no report entries.

SPECS stops executing algorithms as soon as they fail. Pursuing an algorithm beyond the point at which it fails is likely to produce misleading results.

Many of the validation algorithms use "words." A word is a series of one or more non-blank characters, ending at a blank or at the end of an input line. Words may include numerals and punctuation marks.

C.1 SINGLE-FIELD VALIDATION

Single-field validation serves two purposes: it detects possible errors in the value of an input field, and it converts the information to formats that can be used in specimen-data records. Table 6 summarizes the relationship between input fields, validation algorithms, and specimen-data fields.

Single-field validation fails when an algorithm cannot interpret the information it was given or cannot encode it in a way that SPECS can represent. The information that fails may be unusual rather than incorrect, but must be re-entered before SPECS can store it in a record.

Many of the single-field algorithms recognize that data may be broken into parts before it is entered. The validation for "count" is an example: it recognizes that the user may already have distinguished between the count itself and the items that were counted.

TABLE 6

Fields converted by single-field validation

Input field	Specimen-data fields
Accession number	Accession number
Count	Count type, count
Depth	Depth type, depth, perhaps location
Description	Description, description length
Grid square	Grid-square type, N-S coord, L-R half, E-W coord, perhaps location
Location	Location, location length
Site	Site
Specimen number	Accession number, sequence and continuation number, specimen type
Unit	Unit type, unit, perhaps location
Weight	Weight type, weight
Year	Year

C.1.1 Accession number

An accession number is a positive integer with at most four digits. The special value "NONE" is not an accession number, but indicates that there is no accession number.

Validation fails for any value that is not an accession number.

Algorithm:

1. If the value that was given is NONE, validation fails because there is no accession number.

2. If the value is not an integer from 1 to 9999, validation fails because the value is not an accession number.

3. Otherwise, the value is an accession number.

C.1.2 Count

Legal count information is a number, which may be followed by a single blank and a count type telling what was counted. NONE, an indication that there is no count information, is also legal.

Validation fails if a count is not numeric. Although numbers other than non-negative integers are unlikely to be correct counts, they do not cause validation to fail. Non-integer counts are rounded to integers, and counts outside the range -32,768 to 32,767 are re-encoded as -32,768 or 32,767.

SPECS recognizes seven kinds of things that can be counted: bags, jars, boxes, vials, pieces, mixed items (vials and jars, for example), and other. It uses the count type to identify what was counted, then encodes the information as one of the single-character "type codes" shown in Table 7. Note that counts without count types are assumed to be in pieces.

If containers (bags, jars, boxes, or vials) are counted, SPECS compares the count to the "container count" validation

TABLE 7

Count types, designations, and type codes

Units counted	Designations	Type code
Bags	Bag, bags, B	B
Jars	Jar, jars, J	J
Boxes	Box, boxes, X	X
Vials	Vial, vials, V	V
Mixed items	Mixed, M	M
Pieces	Piece, pieces, P, no designation	P
No count	None	N
Other	Anything else	O

constant to find out whether the specimen has implausibly many containers.

SPECS stores the type code in the "count type" field of a master-file or transaction-file record. If there is a count, it is stored in the "count" field.

Algorithm:

1. If the count and count type are unknown, find them.

    1.1 If the count information is NONE, consider it the count type.

    1.2 Otherwise, consider the first word of the count information to be the count, and the rest (if any) to be the count type.

2. Find the count type in the "Designations" column of Table 7, and remember the appropriate type code.

3. Check the format and value of the count.

   3.1 If the count is not a number, validation fails because the count is malformed.

   3.2 If the count is > 32,767, re-encode it as 32,767.

   3.3 If the count is < -32,768, re-encode it as -32,768.

   3.4 If the count is not an integer, round it to the closest integer.

4. If the items counted are containers (bags, jars, boxes, or vials) and the count is larger than the current value of the "container count" validation constant, the container count is too large.

C.1.3 Depth

Legal depth information begins with an entry from the "Designation" column of Table 8 to indicate the depth type. SPECS represents the depth type by the single-character "type code" shown in the table.

If the depth type is level or zone, it must be followed by a single blank, then by a level or zone number. Level and zone numbers are positive integers or positive integers

TABLE 8

Depth types, designations, and type codes

Depth type	Designations	Type code
Level	Level, L, L., Lev, Lev.	L
Zone	Zone, Z, Z.	Z
Surface	Surface, S	S
No depth	None	N
Other	Anything else	O

followed by single letters. SPECS represents these numbers as two digits (of which the leftmost may be a zero) followed by either a blank or a letter. This representation ensures that level and zone numbers can be lexically sorted into order of increasing depth.

SPECS compares level and zone numbers to the current value of the "maximum level" validation constant to find out whether they represent implausibly great depths.

SPECS stores the type code in the "depth type" field of a master-file or transaction-file record. Level and zone numbers are stored in the "depth" field. If the depth type is "other," the "depth type" and "depth" fields do not fully represent the depth information, so SPECS adds the data that the user gave to the "location" field.

Algorithm:

1. Identify the depth type and depth measurement.

1.1 Find the first word of the depth information in table 8, and get the corresponding type code.

1.2 If there is more depth information, assume it is the depth measurement.

2. If the depth type is not "no depth" or "other", check the consistency of the depth type and depth measurement.

   2.1 If the depth type is "surface", there should be no depth measurement. If there is, the surface information cannot be represented.

   2.2 Otherwise, the depth type is level or zone. If the depth measurement is not a positive integer or a positive integer followed by a single letter, the level or zone number is malformed.

3. If the depth measurement has 4 or more characters, truncate it to three characters.

4. If the depth type is level or zone, right-justify the numeric part of the depth measurement as two characters. Concatenate the letter in the depth measurement, or a blank if there is no letter, to the end of the numeric part of the measurement.

5. If the depth type is level or zone and the depth measurement is lexically greater than the current

value of the "maximum level" validation constant, the level or zone number is too high.

C.1.4 Description

Validation of description information is limited to counting the characters in the information and standardizing the spellings of "disk" and "CSPP". Any value is legal for "description," although those with more than 127 characters are truncated. The special value NONE indicates that there is no description information.

The validation algorithm replaces "disc" and "discs" by "disk" and "disks", and replaces "C.S.P.P.", "C. S. P. P.", and "CS PP" by "CSPP".

SPECS stores the description and its length in the "description" and "description length" fields of a master-file or transaction-file record.

Algorithm:

1. If the description is NONE, there is no description. In this case, delete the NONE, remember that the description's length is 0, and stop validation.

2. If the description includes "disc" or "discs" at the end of a word or followed by a punctuation mark, substitute "disk" or "disks".

3. If the description includes "C.S.P.P", "C. S. P. P", or "CS PP", substitute "CSPP".

4. Compute the length of the description by counting the characters in it.

5. If the length is > 127, truncate the description to 127 characters and remember the new length of 127.

C.1.5 Grid square

SPECS recognizes five kinds of legal grid-square information: point proveniences, grid squares, other kinds of areas denoted by ranges of coordinates, other grid-square information, and no information.

Information about point proveniences, grid squares, and areas denoted by ranges has three parts: a N - S coordinate designating a position relative to the east-west axis of the coordinate system, an "L" or "R" indicating a half-site, and an E - W coordinate giving a distance from the north-south axis. E - W coordinates are always non-negative; N - S coordinates may not be.

Each E - W or N - S coordinate must be an integer (typed without a decimal point), a real number (which must include a decimal point), or a range (two integers or real numbers separated by a hyphen). The coordinates for a single grid-square value may be of different types.

SPECS uses the types of the coordinates to distinguish grid squares, point proveniences, and areas. Grid-squares have two integer coordinates, point proveniences have at least one real-number coordinate, and areas have either two range coordinates or a range and an integer. If the grid-square information represents a point provenience, SPECS tries to derive the integer coordinates of the southeast corner of the one-foot square containing the point. When given an area, SPECS derives the southeast corner of the one-foot square containing the southeast corner of the area.

SPECS represents the integer coordinates and "L" or "R" of grid-squares, point proveniences, and areas in the "N-S coord," "L-R half," and "E-W coord" fields of master-file and transaction-file records, and encodes the kind of grid square information in the "grid-square type" field. These fields do not represent all of the information given by point proveniences, areas, and other grid-square information, so SPECS adds the original grid-square data for these types to the "location" field.

The validation algorithm confirms that the coordinates are within the ranges set for them by the "maximum E-W for "L"," "maximum E-W for "R"," "minimum E-W for "L"," "minimum E-W for "R"," "minimum N-S," and "maximum N-S" validation constants.

<u>Algorithm</u>:

1. If SPECS does not already know what kind of grid-square information was given, find out.

    1.1 If the grid square information is NONE, there is no grid square. Encode the grid square type as N, and stop validation.

1.2 Find the first "L" or "R" in the grid square information. If an L is found, the grid-square information represents a location in the left half of the site; if an R is found, the location is in the right half. Consider everything preceding the L or R to be a north-south coordinate and everything following it to be an east-west coordinate. If there is no "L" or "R", validation fails because the grid square is malformed.

1.3 For each of the east-west and north-south coordinates, find out whether it is:

a) an integer

b) a real number, or

c) a range.

If a coordinate does not fall into one of these categories, validation fails because the coordinate is malformed.

1.4 If both coordinates are integers, the grid-square information represents a point with integer coordinates.

1.5 If at least one of the coordinates is a real number, the grid-square information represents a point provenience.

   1.6 If the grid square is not a point provenience and at least one of the coordinates is a range, the grid-square information represents an area.

   1.7 Encode the grid-square type. If the grid square is a point with integer coordinates, its type is G; if it is a point provenience, its type is P; if it is an area, its type is A.

2. If the east-west coordinate of an integer or point provenience or a number in a range of east-west coordinates is negative, it is probably an error.

3. Convert each real-number coordinate to an integer.

   3.1 Convert an east-west coordinate in the left half of the site to an integer by taking its floor (the greatest integer <= the coordinate).

   3.2 Convert an east-west coordinate in the right half of the site to an integer by taking its ceiling (the smallest integer >= the coordinate).

   3.3 Convert a north-south coordinate to an integer by taking its floor.

4. Convert each range coordinate to an integer representing its south or east end.

4.1 Convert a range of east-west coordinates in the left half of the site to an integer by taking the floor of the lower number of the range.

4.2 Convert a range of east-west coordinates in the right half of the site to an integer by taking the ceiling of the higher number of the range.

4.3 Convert a north-south coordinate to an integer by taking the floor of the algebraically smaller number of the range.

5. Confirm that the coordinates are within the ranges allowed by the validation constants.

5.1 If the grid square is in the left half of the site and the east-west coordinate is not between the values of "minimum E-W for "L"" and "maximum E-W for "L"", the east-west coordinate for the left half of the site is not within the expected range.

5.2 If the grid square is in the right half of the site and the east-west coordinate is not between the values of "minimum E-W for "R"" and "maximum E-W for "R"", the east-west coordinate for the right half of the site is not within the expected range.

5.3 If the north-south coordinate is not between the values of "minimum N-S" and "maximum N-S", the north-south coordinate is not within the expected range.

C.1.6 Location

Validation of location information is limited to computing and checking its length. No attempt is made to standardize location information. Anything is legal as a value for "location," although values with more than 127 characters are truncated. The special value NONE indicates that there is no location information.

Algorithm:

1. If the location information is NONE, there is no location. In this case, delete the NONE, remember that the location's length is 0, and stop validation.

2. Compute the length of the location information by counting the characters in it.

3. If the length is > 127, truncate the location information to 127 characters and remember the new length of 127.

C.1.7 Site number

A site number has one to nine characters. It is often composed of a two-letter county designation, a one-letter code indicating a hypothesized site function, one to five digits of sequence number, and a letter.

The special value NONE is recognized as an indication that there is no site information, but is considered an error. No other checking of the contents of a site number is performed.

Algorithm:

1. If the site information is NONE, validation fails because the site number is missing.

2. If there is not at least one character in the site number, validation fails because the site number is missing.

3. If there are more than 9 characters in the site number, validation fails because the site number is too long.

C.1.8 Specimen number

A specimen number is entered as a string of 2 to 13 characters. As was described in Chapter 2, a specimen number is composed of an accession number, a specimen type code, a sequence number, and an optional slash and continuation number. Specimen numbers for new specimens may sometimes be entered without accession numbers, as discussed in Appendix A.

The special value NONE is recognized as an indication that there is no specimen number but, because SPECS cannot represent a specimen without a number, is considered erroneous.

The validation algorithm changes two-letter specimen type codes into the one-letter codes that SPECS can represent. The codes "Bu" and "hb" are both changed to "h," and "eb" is changed to "e."

SPECS encodes the sequence number as the leftmost four digits of a six-digit "sequence and continuation number," and the continuation number as the rightmost two digits. Sequence numbers and continuation numbers are right-justified, and may have leading zeros to give them the appropriate numbers of digits. Specimen numbers without continuation numbers are represented with continuation numbers of 00.

Specimen-number information is stored in the "accession number," "specimen type," and "sequence and continuation number" fields of master-file and transaction-file records.

Algorithm:

1. If the specimen number has less than 2 or more than 13 characters, validation fails because the specimen number is malformed.

2. If the specimen number is NONE, validation fails because there is no specimen number.

3. Break the specimen-number information into its constituent parts:
    i) The accession number, a series of 0 or more digits preceding the first non-digit character.

ii) The specimen type, a sequence of at least one non-digit character.

iii) The sequence number, a series of one or more digits, ending at a "/" or the end of the specimen number.

iv) (Optional) The continuation number, a "/" followed by a series of digits.

If this cannot be done, validation fails because the specimen number is malformed.

4. Check the format of the accession number.

4.1 If it has 0 digits, no accession number was given; the accession number is correctly formatted.

4.2 If it is a 1 - 4 digit positive integer, the accession number is correctly formatted.

4.3 Otherwise, validation fails because the accession number is malformed.

5. Check the format of the specimen type, and recode it as needed.

5.1 The specimen type must be "a", "b", "c", "e", "eb", "h", "hb", "Bu", "m", or "p". If it is not, validation fails because the specimen type is malformed.

5.2 If the specimen type is "Bu" or "hb", change it to "h".

5.3 If the specimen type is "eb", change it to "e".

6. Check the format of the sequence number.

6.1 If it is a 1 - 4 digit positive integer, it is correctly formatted. Add leading zeros as needed to produce four digits.

6.2 Otherwise, validation fails because the specimen number is malformed.

7. Check the format of the continuation number.

7.1 If no continuation number was given, assume it is 00.

7.2 If the continuation number is a "/" followed by a 1 - 2 digit integer, ignore the "/." Add a leading zero to the integer, if needed, to produce two digits.

7.3 Otherwise, validation fails because the continuation number is malformed.

8. Concatenate the continuation number to the end of the sequence number, producing a 6-digit "sequence and continuation number."

C.1.9 Unit

Legal unit information begins with an entry from the the "Designations" column of Table 9 to indicate a unit type. As the table shows, SPECS recognizes seven unit types: features, burials, post holes, test pits, structures, no unit information, and other. SPECS encodes the unit type as a single-character "type code."

TABLE 9

Unit types, designations, and type codes

Unit type	Designations	Type code
Feature	Feature, F, F., Fea, Fea.	F
Burial	Burial, B, Bur, Bur.	B
Post hole	Post hole, Posthole, PH, P.H., P. H., P	P
Test pit	Test pit, pit, T	T
Structure	Structure, struc., S	S
No unit	None	N
Other	Anything else	O

If the unit type is not "no unit" or "other," it must be followed by a single blank and a unit number. Unit numbers are either positive integers less than 32,768 or such integers followed immediately by single letters.

Unit types are represented in the "unit type" fields of master-file and transaction-file records. If the unit type is not "no unit" or "other," the integer part of the unit number is stored in the "unit" field. For units of type "other," SPECS adds the unit information to the "location" field.

Units whose numbers include letters (e.g. Burial 49A) are rare. They are identified in two places: by the integer parts of their numbers in the "unit type" and "unit" fields (as Burial 49, for example), and by an abbreviation of their full designations (such as B 49A) in the "location" field. The abbreviation is formed by concatenating the unit type code and unit number, separated by a single blank.

<u>Algorithm</u>:

1. If the unit type and number are already known, look the type up in the "Designations" column of Table 9 to find its type code.

2. If the unit type and number are unknown, find them.

    2.1 If the unit information is NONE, consider it the unit type. Its type code is N.

    2.2 Otherwise, find the first word or two of the unit information in the "Designations" column of Table 9, and remember the type code shown in the table.

2.3 If the type code is not N or O, consider the rest of the unit information to be the unit number.

3. If the unit type is N or O, consider the unit information correct and stop validation.

4. If the unit number is not a positive integer <= 32,767, or such an integer followed by a single letter, validation fails because the unit number is malformed.

5. If the unit number includes a letter, create an abbreviated form of the unit information by concatenating, in left-to-right order, the unit type code, a single blank, and the unit number.

C.1.10 Weight

Legal weight information is either NONE or a non-negative number representing a weight in ounces. "<" and ">" signs may precede the weight, as they do in the specimen catalog, to indicate that the recorded values are higher or lower than the true weights.

Weights may be followed by "oz" or "oz." to label them as in ounces. The validation algorithm considers all other labels erroneous, but assumes that weights without labels are in ounces.

SPECS stores weight information in the "weight type" and "weight" fields of master-file and transaction-file records.

The "weight type" field has one of four values: N if there is no information, G if the true weight is greater than the recorded weight (the user entered a ">"), L if the true weight is less than the recorded weight (the user entered a "<"), or E if the user gave no indication that the true weight and recorded weight are not equal. If the weight type is not N, SPECS converts the weight to tenths of ounces and stores it in the "weight" field.

<u>Algorithm</u>:

1. If the weight information is NONE, encode the weight type as N.

2. Otherwise, break the weight information into its component parts.

   2.1 Find the weight measurement, a non-negative integer or real number at the end of a word. If there is none, validation fails because the weight is malformed.

   2.2 If the weight measurement is preceded by a single "<," encode the weight type as L.

   2.3 If the weight measurement is preceded by a single ">," encode the weight type as G.

   2.4 If the weight measurement is preceded by anything other than "<" or ">," validation fails because the weight is malformed.

   2.5 If the weight measurement is followed by anything other than one or more blanks and a designation for ounces ("oz" or "oz."), validation fails because the weight is not in ounces.

   2.6 If the weight type is still unknown, encode it as E.

3. Alter the representation of the weight measurement as needed.

   3.1 Multiply the weight measurement by 10. (SPECS expects to divide weights by 10 to get ounces.)

   3.2 If the weight measurement is not an integer, round it to one.

   3.3 If the weight measurement is > 32767, re-encode it as 32767 and set the weight type to G.

C.1.11 Year

A year is entered as a 4-digit integer. The special value NONE is recognized as an indication that there is no year information, but is considered an error.

Algorithm:

1. If the year information is NONE, validation fails because the year is missing.

2. If the year is not a non-negative 4-digit integer, validation fails because the year is incorrectly formatted.

3. If the year is not greater than 1937, it is too early.

4. If the year is greater than the present year, it is too late.

C.2 MULTIPLE-FIELD VALIDATION

Multiple-field validation checks the consistency of the values of several fields for the same specimen. In multiple-field validation, failure means that the values appear inconsistent, not that the user must change them. Correct combinations of values may fail multiple-field validation because they are unusual or have not been entered before.

SPECS uses three multiple-field validation algorithms: type-description, kind-count-weight, and site-yr-acc.

C.2.1 Type-description validation

Type-description validation checks the consistency of specimen types and descriptions, using the principle that some words in descriptions are applied almost exclusively to specimens of a single type. If these words are used with specimens of another type, validation fails because the specimen type and description are inconsistent. Specimens whose descriptions do not contain words used by the algorithm are assumed to be correct, as are comments and specimens in which the words occur with the expected types.

The reliability of the algorithm clearly depends on the selection of the words that are used. In general, the words chosen for the algorithm meet two criteria: 1) they occur in at least 1% of all of the descriptions of the specimens listed in Appendix G, and 2) in at least 80% of the descriptions in which they occur, the words are applied to specimens of a single type. These words, and the types with which they are associated, are shown in table 10.

Algorithm:

1. If the specimen type is "c," consider it consistent with the description and stop validation.

2. If the description has a word listed in table 10 and the specimen type is not the type listed in the table, validation fails because the description and specimen type are inconsistent.

TABLE 10

Specimen types and words from descriptions

Words from descriptions	Types
"blade", "blades"	a
"CSPP"	a
Any word ending in "er", "ers", or "ed"	a
"pipe", "pipes"	a
"clay"	a
"disk", "disks"	a
"bead", "beads"	a
"chip", "chips"	m
"flake", "flakes"	m
"soil sample"	m
"daub"	m
"raw material"	m
"washing"	m
"animal"	b
"bone", "bones"	b
"fauna"	b
"charcoal"	e
any word containing "sherd"	p

C.2.2 Kind-count-weight validation

Kind-count-weight validation uses specimen types, descriptions, counts, and weights to decide whether the number and size of pieces in a specimen are consistent with the current values of the validation constants listed in Appendix B.

The algorithm begins by using the specimen type and description to assign the specimen to a "kind." All specimens in a kind are assumed to have roughly the same number of pieces of roughly the same sizes. Table 11 shows the relationship between specimen types, descriptions, and kinds.

The weights of soil samples and washings, for which pieces are not counted, are compared to the maximum weights for them in the validation constants. Validation fails if the specimens are too heavy.

The numbers of pieces in specimens for which pieces are counted are compared to the validation constants giving maximum counts for specimens of the appropriate kinds. Validation fails if the specimens have too many pieces.

The validation algorithm computes the numbers of pieces per tenth of an ounce for specimens for which pieces were counted and non-zero weights were recorded. These numbers, called "densities," must fall between the values of the validation constants giving minimum and maximum densities for specimens of the appropriate kind. If the density for

TABLE 11

Types and descriptions for kinds of specimens

Type	Description	Kind
a	Anything with "bead"	Beads
a	Anything without "bead"	Miscellaneous type "a" specimens
b	Anything	Type "b" specimens
e	Anything	Type "e" specimens
h	Anything	Type "h" specimens
m	Anything with "flake" or "chip"	Flakes and chips
m	Anything with "soil sample"	Soil samples
m	Anything with "washing"	Washings
m	Anything without "flake", "chip", "soil sample", or "washing"	Miscellaneous type "m" specimens
p	Anything	Potsherds

the specimen is outside the limits set by the validation constants, validation fails.

Validation succeeds for all comments and for other specimens for which it is not known to fail.

Algorithm:

1. If the specimen is a comment (type "c"), consider it

correct. Otherwise, use the specimen type and description to find the appropriate kind from Table 11.

2. If the kind is soil sample, check its weight.

   2.1 If there is no weight, or if the weight is <= the maximum soil sample weight given by the "soil-sample weight" validation constant, consider the specimen correct.

   2.2 Otherwise, validation fails because the soil sample is too heavy.

3. If the kind is washings, check its weight.

   3.1 If there is no weight, or if the weight is <= the maximum washings weight given by the "washings weight" validation constant, consider the specimen correct.

   3.2 Otherwise, validation fails because the washings are too heavy.

4. If the kind is not soil samples or washings, check its count.

   4.1 If there is no count, or the count is not in pieces, consider the specimen correct.

4.2 Otherwise, if the count is greater than the maximum count given by the validation constants for the kind, validation fails because the count is unexpectedly large.

5. If the kind is anything but soil samples or washings, the count is in pieces, and the weight is given and > 0, check the specimen's density.

5.1 Compute density = count / weight.

5.2 If the density is not between the minimum and maximum densities given by the validation constants for the kind, validation fails because the density is outside the expected range.

C.2.3 Site-yr-acc validation

SPECS uses site-yr-acc validation to check the consistency of information about sites, years, and accession numbers. If the site, year, and accession number are not in the site-yr-acc file, validation fails.

Algorithm:

1. Generate the key for the site-yr-acc file by concatenating the year to the site.

2. Look for the key in the site-yr-acc file, retrieving all records that have the key.

2.1 If the key is not in the file, validation fails because the site and year are not entered in the file.

2.2 If the key is in the file with the accession number being checked, the combination of site, year, and accession number is consistent.

2.3 Otherwise, validation fails because the accession number is inconsistent with previous entries in the site-yr-acc file.

C.3 SEQUENCE CHECKING

Sequence checking is performed for pairs of specimens with sequential specimen numbers. SPECS uses two kinds of sequence checking: for specimen numbers and for depths. Sequence checking for specimen numbers confirms that the specimen numbers are indeed sequential. Sequence checking for depths confirms that specimens collected at a single location and assigned sequential specimen numbers are recorded in increasing order of depth. Both kinds of validation are intended to detect typographical errors in specimen numbers or location information.

Data fails sequence checking because it is inconsistent with SPECS' assumptions about the order in which specimens are assigned specimen numbers and entered, and not because it is erroneous or because SPECS cannot represent it. The user is not required to change information that fails sequence checking.

Specimen numbers are considered sequential when one of the following conditions is true:

1. Neither of the specimen numbers has a continuation number, and the second number to be entered is one greater than the first. That is, 2002a103 and 2002a104 are sequential, but 2002a103 and 2002a105 are not.
2. Both specimens have the same sequence number, the first specimen number has no continuation number, and the second has a continuation of /1. Thus, 2002a103 and 2002a103/1 are sequential. 2002a103 and 2002a103/2 are not sequential, nor are 2002a103 and 2002a104/1.
3. Both specimen numbers have continuation numbers, their sequence numbers are the same, and the continuation number of the second specimen is one greater than that of the first. Thus, 2002a103/1 and 2002a103/2 are sequential. Neither 2002a101/1 and 2002a101/3 nor 2002a101/1 and 2002a102/1 are sequential.
4. The first specimen has a continuation number, the second does not, and the sequence number of the second is one greater than the sequence number of the first. Thus, 2002a103/1 and 2002a104 are sequential.

Neither 2002a103/1 and 2002a105 nor 2002a103/1 and 2002a103 are sequential.

C.3.1 Sequence checking for specimen numbers

Sequence checking for specimen numbers is used exclusively during interactive sessions. It compares each specimen to the specimen most recently entered with the same INPUT command, and fails if their specimen numbers are not sequential. Sequence checking for specimens is most useful when information about a series of specimens is being retyped from a Specimen Catalog form.

Three conditions cause a pair of specimen numbers to fail validation:

1. Gaps in the specimen-number sequence, which may be caused by the user's mistyping a specimen number or accidentally skipping a specimen when he is entering data from a Specimen Catalog form. Gaps also occur in correct data, of course, if the user enters information that is not in sequence.
2. Duplicate specimen numbers, which may be caused by the user's mistyping a specimen number or accidentally re-entering a specimen. Duplicate numbers may occur in correct data when subspecimens of the same specimen are entered.
3. Overlaps in the specimen-number sequence, which may be caused by the user's mistyping a specimen number

or accidentally re-entering specimens. Overlaps may occur in correct data when the user enters information that is not in sequence.

Algorithm:

(In this algorithm, the first and second sequence numbers derive their names from the order in which they were entered. "Sequence number" is used as an abbreviated name for the "sequence and continuation number" produced by single-field validation for specimen numbers.)

1. If the accession numbers for the specimens are not the same, assume that the specimens are entered in sequence, and stop validation.

2. Compute the "spread" of the sequence numbers by subtracting the first from the second.

3. If the spread is negative, the second specimen number is lower than the first. In this case, validation fails because the specimen numbers overlap.

4. If the spread is 0, the first and second specimen numbers are the same. In this case, validation fails because duplicate specimen numbers were entered.

5. If the spread is positive, examine the sequence numbers more closely.

5.1 If both sequence numbers are multiples of 100, neither specimen number has a continuation. In this case, the spread must be 100 if the specimens are in sequence (condition 1 in the definition of sequential specimens). If the spread is greater than 100, validation fails because there is a gap in the specimen-number sequence.

5.2 If the second sequence number is not a multiple of 100, one or both of the specimen numbers have continuations. In this case, the spread must be 1 if the specimen numbers are in sequence (conditions 2 and 3 in the definition of sequential specimens). If the spread is greater, validation fails because there is a gap in the specimen-number sequence.

5.3 If the second sequence number is a multiple of 100 and the first is not, only the first specimen number has a continuation. In this case, the spread must be less than 100 if the specimen numbers are in sequence (condition 4 in the definition of sequential specimens). If the spread is 100 or more, validation fails because there is a gap in the specimen number sequence.

C.3.2 Sequence checking for depths

Sequence checking for depths is performed during interactive sessions and batch updates. During interactive sessions, it is limited to comparing each specimen with the last specimen entered with the same INPUT command. Like sequence checking for specimen numbers, sequence checking for depths is most useful when information about a series of specimens is being retyped from a Specimen Catalog form.

In sequence checking for depths, SPECS confirms that the depth of the second specimen of a pair collected from the same site, year, unit, and grid square and assigned sequential specimen numbers is at least as great as that of the first.

Depth comparisons follow three rules:

1. Any depth is at least as deep as the surface, and any level or zone is deeper.
2. Levels and zones cannot be mixed for specimens with sequential numbers that were collected at the same location. That is, if one specimen was collected from a level, the other must not have been collected from a zone. (Levels within zones, an apparent exception to this rule, are excluded from consideration by the representation of depth information described in Appendix A.)
3. One level or zone is deeper than another if its level or zone number is lexically greater.

Sequence checking for depth will fail under these circumstances:

1. A mistyped specimen number, site, year, unit, or grid square may cause specimens that are not sequentially numbered and from the same location to appear to be. This kind of error may result from overly free use of ditto marks in entering specimen numbers and location information, as well as from the more usual sorts of typographical errors.
2. The specimens from each location that were assigned sequential specimen numbers were not sorted by depth during laboratory processing. In this case, the data may be correct.
3. Levels and zones are mixed for specimens with sequential numbers that were collected at the same location. In this case, the data may be correct.
4. Level and zone numbers are not in nondecreasing lexical order. In this case, the data may be correct.

Algorithm:

(In this algorithm, the "first specimen" is the one with the lower specimen number, regardless of the order in which the specimens were entered.)

1. If the specimens do not have sequential numbers, assume that their depths are consistent, and stop

validation. (The algorithm for sequence checking for specimen numbers may be used to determine whether the specimen numbers are sequential.)

2. If the specimens are not from the same site, year, unit, and grid square, assume the depths are consistent and stop validation.

3. If the second specimen is from the surface and the first is from a level or zone, validation fails.

4. If one specimen is from a level and the other is from a zone, validation fails.

5. If both specimens are from levels or both from zones, validation fails if the level or zone number of the first is lexically greater than that of the second. Otherwise, validation succeeds.

Appendix D

REPORTS

This appendix lists the report entries and terminal reports that SPECS can produce. It has four sections: session-report entries, terminal reports, update-report entries, and an alphabetized list of values to be substituted into the reports when they are generated.

Three notational conventions used in the reports are:

1. Anything that is underlined will be replaced by an appropriate value when the report entry containing it is generated. Anything else is to be printed as it appears in the report listings.
2. Anything that is enclosed in square brackets may be omitted under some circumstances. These circumstances are listed for each entry in which square brackets appear.
3. Each session report entry is preceded by "SRE" and a number. Similarly, each terminal report is preceded by "Terminal Report," and each update-report entry is preceded by "URE." These are cross-references from the interactive-session and batch-update pseudocode (Appendices E and F), and should not be printed in the reports.

Terminal Report 4 should be printed as it appears in this appendix, but the other report entries and terminal reports may be reformatted by the implementor.

D.1 SESSION-REPORT ENTRIES

SRE 1:

SESSION REPORT

SPECS Version 1

date, time

SRE 2:

The session ended because the user refused to give his name.

SRE 3:

The user is: username.

SRE 4:

The session ended because SPECS could not open the system parameters file.

SRE 5:

SPECS could not read the progress record from the system parameters file.

SRE 6:

SPECS could not read the validation-constants record from the system parameters file.

SRE 7:

SPECS could not read the passwords record from the system parameters file.

SRE 8:

The session ended because SPECS could not open the system parameters file to write new values in it.

SRE 9:

System parameters file reconstruction was attempted, but failed when the user refused to supply the parameters password.

SRE 10:

System parameters file reconstruction was attempted, but failed when the user refused to supply the password password.

SRE 11:

System parameters file reconstruction was attempted, but failed when the user refused to supply the constant validation constant.

SRE 12:

System parameters file reconstruction was attempted, but failed when the user refused to give the identification of the current master and transaction files.

SRE 13:

System parameters file reconstruction was attempted, but failed when the user refused to give the next transaction number to be assigned.

SRE 14:

System parameters file reconstruction was attempted, but failed when the user refused to provide progress information whose correctness he would confirm.

SRE 15:

System parameters file reconstruction was attempted, but failed when SPECS could not write the record in the file.

SRE 16:

The system parameters file was reconstructed.

SRE 17:

The session is to reset the information needed to reconstruct a master file.

The file identification is now file ID.
The next transaction to be assigned is next trans#.

SRE 18:

The session ended because the user refused to specify whether he wanted to reset the values needed for master file reconstruction.

SRE 19:

The information was not reset because the user failed to provide the master password.

SRE 20:

The information was not reset because the user refused to identify the master and transaction files to be used in the next batch update.

SRE 21:

The information was not reset because the user refused to identify the next transaction number to be assigned.

SRE 22:

The new values are:

File identification: file ID.

Next transaction number: next trans#.

SRE 23:

The session ended because SPECS could not open the site-yr-acc file.

SRE 24:

The session ended because SPECS could not open the current transaction file for reading.

SRE 25:

The session ended because SPECS could not read the first record on the transaction file.

SRE 26:

The session ended because the file provided for transactions was not a transaction file.

SRE 27:

The session ended because the transaction file was not current.

SRE 28:

The session ended because the transaction file is a multiple-update file. It contains transaction files with identifications ID list.

SRE 29:

The session ended because, although the current transaction file was given, it could not be opened for writing.

SRE 30:

The transaction file for this session is: file ID.
It was created file date.

SRE 31:

SPECS ignored a command command because the user failed to supply the correct password.

SRE 32:

Command:

CANCEL TRANSACTION cancel# FOR spec#

Response:

SRE 33:

Transaction # trans# was written, with these values:

Accession number: acc#

Sequence and continuation number: <u>seq#</u>

Specimen type: <u>type</u>

Transaction number: <u>trans#</u>

Transaction type code: X

Affected transaction number: <u>cancel#</u>

Transaction # <u>cancel#</u> will be cancelled during the next batch update.

SRE 34:

Command:

CHANGE [ALL] <u>field</u> FOR <u>spec#</u> [WITH "<u>description</u>"]

NEW VALUE: <u>value</u>

(In this entry, [ALL] appears only if ALL was part of the original command. [WITH "<u>description</u>"] appears only if description information was given.)

Response:

SRE 35:

The command was not executed because the new count information was not numeric.

SRE 36:

The new count was negative, and is probably an error.

SRE 37:

The new count information was not an integer. It was rounded to <u>count</u>.

SRE 38:

The new count was too far negative to represent, and has been changed to -32,768. This count is probably erroneous.

SRE 39:

The new count was too large to represent, and has been changed to 32,767. This count is probably erroneous.

SRE 40:

The number of boxes, bags, vials, or jars was larger than <u>container count</u>, the maximum number of containers allowed by the validation constants.

SRE 41:

SPECS ignored information about the depth of a specimen from the surface.

SRE 42:

SPECS did not recognize the information it was given as a level or zone number, but treated it as one anyway.

SRE 43:

The depth information was truncated to 3 characters.

SRE 44:

The new level or zone number was higher than <u>max level</u>, the maximum level or zone number allowed by the validation constants.

SRE 45:

"Disc" was changed to "disk."

SRE 46:

"Old CSPP" was changed to "CSPP."

SRE 47:

The new description information was truncated to 127 characters.

SRE 48:

The command was not executed because the user refused to tell SPECS whether the new information should replace, or be added to, the old.

SRE 49:

SPECS could not interpret the new information it was given. The information may be erroneous.

SRE 50:

The command was not executed because the user refused to break the grid square information into parts.

SRE 51:

There is no new grid square information.

SRE 52:

The new grid square information represents an area.

SRE 53:

The new grid square information represents something other than a grid square, point provenience, or area.

SRE 54:

The new grid square information represents a grid square.

SRE 55:

The new grid square information represents a point provenience.

SRE 56:

The east-west coordinate is negative, and is probably an error.

SRE 57:

The east-west coordinate was converted to an integer.

SRE 58:

The north-south coordinate was converted to an integer.

SRE 59: The north-south and east-west coordinates were converted to integers representing the southeast corner of the area.

SRE 60:

The east-west coordinate is not between L min E-W for L and L max E-W for L, as the validation constants indicate that it should be.

(This entry is printed with no spaces between the "L" and the value to be filled in.)

SRE 61:

The east-west coordinate is not between R <u>min</u> <u>E-W</u> <u>for</u> <u>R</u> and R <u>max</u> <u>E-W</u> <u>for</u> <u>R</u>, as the validation constants indicate that it should be.

(This entry is printed with no spaces between the "R" and the value to be filled in.)

SRE 62:

The north-south coordinate is not between <u>min</u> <u>N-S</u> and <u>max</u> <u>N-S</u>, as the validation constants indicate that it should be.

SRE 63:

The new location information was truncated to 127 characters.

SRE 64:

The command was not executed because it required that the specimen be recorded without a site.

SRE 65:

The command was not executed because the new value has too many characters to be a site designation.

SRE 66:

The command was not executed because SPECS could not interpret the new value as a specimen number.

SRE 67:

The command was not executed because SPECS did not know the accession number for the new specimen number.

SRE 68:

Specimen types "Bu" and "hb" are recorded as "h."

SRE 69:

Specimen type "eb" is recorded as "e."

SRE 70:

The command was not executed because SPECS cannot represent a unit number that is neither a positive integer nor a positive integer followed by a single letter.

SRE 71:

An abbreviation of the unit information was created, and will be added to the "location" information.

SRE 72:

The command was not executed because SPECS could not interpret the information as a weight in ounces.

SRE 73:

The weight was rounded to the closest 0.1 ounce.

SRE 74: The weight was too large to represent, and was recorded as being greater than 3276.7 ounces (almost 205 pounds). This weight is probably erroneous.

SRE 75:

The command was not executed because it required that the specimen be recorded without a year.

SRE 76:

The command was not executed because the year information was not a non-negative four-digit integer.

SRE 77:

The year is too early to be correct for any of the current RLA collections, and is probably erroneous.

SRE 78:

The year has not yet arrived.

SRE 79:

Transaction # <u>trans#</u> was written, with these values:

SRE 80:

Accession number: <u>acc#</u>

Sequence and continuation number: <u>seq#</u>

Specimen type: <u>type</u>

Transaction number: <u>trans#</u>

Transaction type code: C

Transaction options: <u>options</u>

Input field: <u>field</u>

Field count: <u>field count</u>

SRE 81:

Description: <u>description</u>

SRE 82:

Field designation:

field designation (field changed)

SRE 83:

Field size: field size

SRE 84:

Field format: field format

SRE 85:

New value: new value

SRE 86:

At the next batch update, information from the field field will be changed for [all] specimens numbered spec# [that have description "description"].

(In this entry, [all] appears only if ALL was part of the CHANGE command, and [that have description "description"] appears only if description information was given in the command.)

SRE 87:

After TWO batch updates, the specimen number will be changed for [all] specimens currently numbered spec# [that have description "description"].

(In this entry, [all] appears only if ALL was part of the CHANGE command, and [that have description "description"] appears only if description information was given in the command.)

SRE 88:

Command:

```
CHECK TRANSACTION check# FOR spec#
```

Response:

SRE 89:

Transaction # trans# was written, with these values:

Accession number: acc#

Sequence and continuation number: seq#

Specimen type: type

Transaction number: trans#

Transaction type code: R

Affected transaction number: check#

In the next batch update, validation will be restored for transaction # check#.

SRE 90:

USER'S COMMENT: comment

SRE 91:

Command:

```
CONFIRM TRANSACTION confirm# FOR spec#
```

Response:

SRE 92:

Transaction # trans# was written, with these values:

Accession number: acc#

Sequence and continuation number: seq#

Specimen type: type

Transaction number: trans#

Transaction type code: 0

Affected transaction number: confirm#

In the next batch update, validation will be overridden for transaction # confirm#.

SRE 93:

Command:

```
DELETE [ALL] spec# [WITH "description"]
```

Response:

(In this entry, [ALL] appears only if ALL was part of the DELETE command, and [WITH "description"] appears only if description information was given.)

SRE 94:

Transaction number trans# was written, with these values:

Accession number: acc#

Sequence and continuation number: seq#

Specimen type: type

Transaction number: trans#

Transaction type code: D

Transaction options: options

[Description: description]

The next batch update will delete specimen-catalog information for [all] specimens numbered spec# [that have description "description"].

(In this entry, [all] appears only if ALL was part of the DELETE command. [Description: description] and [that have description "description"] appear only if description information was given.)

SRE 95:

Command execution for this session has ended.

SRE 96:

The field information was copied from specimen old spec#.

SRE 97:

The site, year, and accession number were copied from specimen old spec#.

SRE 98:

A site-yr-acc record was written for site <u>site</u>, year <u>year</u>, and accession number <u>acc#</u>.

SRE 99:

Other accession numbers already recorded for <u>site</u> (<u>year</u>) are <u>acc#</u> <u>list</u>.

SRE 100:

<u>Spec#</u>, the current specimen number, is not in the expected sequence with <u>old</u> <u>spec#</u>, the last specimen number entered.

SRE 101:

Because the north-south and east-west coordinates do not fully represent the grid square, the original grid-sqare information will be added to the "location" field.

SRE 102:

The unit information will be represented both as <u>unit#</u> in the "unit" field of the specimen-data record and in an abbreviated form in the "location" field.

SRE 103:

The unit information for units of type "other" is added to the "location" field of the specimen-data record.

SRE 104:

The depth information for depth measurements of type "other" is added to the "location" field of the specimen-data record.

SRE 105:

The information from this location is not recorded in order of increasing depth. Some of the location information may have been typed incorrectly.

SRE 106:

Most specimens whose descriptions contain <u>keyword</u> are of type <u>type</u>, but this one is not. The specimen type or description may be erroneous.

SRE 107:

The <u>kind</u> weight is greater than <u>max</u> <u>weight</u>, the maximum weight allowed for it by the validation constants.

SRE 108:

The number of pieces in the specimen is greater than <u>max</u> <u>count</u>, the largest number allowed by the validation constants for <u>kind</u>.

SRE 109:

The number of pieces per ounce in the specimen is not between <u>min</u> <u>density</u> and <u>max</u> <u>density</u>, the range allowed by the validation constants for <u>kind</u>.

SRE 110:

Command:

INPUT

New values:

Specimen number: spec#

Site: site

Year: year

Grid square: grid info

Unit: unit info

Depth: depth info

Location: location info

Count: count info

Weight: weight info

Description: description info

Validation messages:

SRE 111:

Response:

Transaction # trans# was written, with these values:

Accession number: acc#

Sequence and continuation number: seq#

Specimen type: type

Transaction number: trans#

Transaction type code: I

Site: site

Year: year

Unit type: unit type

Grid-square type: gs type

Depth type: <u>depth type</u>

Count type: <u>count type</u>

Weight type: <u>weight type</u>

Location length: <u>loc length</u>

Description length: <u>descr length</u>

Unit: <u>unit ID</u>

N-S coord: <u>N-S coord</u>

L-R half: <u>L-R half</u>

E-W coord: <u>E-W coord</u>

Depth: <u>depth</u>

Count: <u>number</u>

Weight: <u>ounces</u>

Location: <u>loc</u>

Description: <u>descr</u>

This information, for specimen <u>spec#</u>, will be added to the master file during the next batch update.

(In this entry, values of fields that do not exist in the record are printed as NONE.)

SRE 112:

Command:

RESET <u>reset name</u>

Response:

SRE 113:

The command was not executed because the user failed to give the correct password.

SRE 114:

The old value of <u>constant</u> is <u>constant</u> <u>value</u>.

SRE 115:

The value of the <u>reset</u> <u>name</u> remains unchanged.

SRE 116:

The <u>password</u> was reset.

SRE 117:

The new value of <u>constant</u> is <u>constant</u> <u>value</u>.

SRE 118:

Command:

```
SET-ACC FOR site(year) TO acc#
```

Response:

SRE 119:

No previous accession numbers are recorded for <u>site</u>(<u>year</u>).

SRE 120:

The command was not executed because the user refused to specify whether the accession number given in the command replaced or supplemented those already in the file.

SRE 121:

Accession number(s) <u>acc#</u> <u>list</u> for <u>site</u>(<u>year</u>) are deleted from the site-yr-acc file.

SRE 122:

Command:

```
SHOW-ACC FOR site(year)
```

Response:

SRE 123:

No accession numbers are recorded for site(year).

SRE 124:

Accession number acc# list is recorded for site(year).

SRE 125:

The user requested a batch update, but failed to supply the update password.

SRE 126:

SYSTEM PARAMETERS REPORT

SRE 127:

The user requested that the passwords be printed, but failed to provide the master password.

SRE 128:

The current passwords are:

Parameters: parameters
Master: master
Input: input
Change: change
Delete: delete
Cancel: cancel

Validation-control: *validation*

Show-acc *show-acc*

Set-acc: *set-acc*

Update: *update*

SRE 129:

The current validation constants are:

Soil sample weight: *soil weight*

Washings weight: *washings weight*

Bead count: *bead count*

Type-"a" count: *a-count*

Type-"b" count: *b-count*

Type-"e" count: *e-count*

Type-"h" count: *h-count*

Flake count: *flake count*

Type-"m" count: m-count

Type-"p" count: *p-count*

Container count: *container count*

Maximum bead density: *max bead density*

Maximum type-"a" density: *max a-density*

Maximum type-"b" density: *max b-density*

Maximum type-"e" density: *max e-density*

Maximum type-"h" density: *max h-density*

Maximum flake density: *max flake density*

Maximum type-"m" density: *max m-density*

Maximum type-"p" density: *max p-density*

Minimum bead density: *min bead density*

Minimum type-"a" density: *min a-density*

Minimum type-"b" density: min b-density

Minimum type-"e" density: min e-density

Minimum type-"h" density: min h-density

Minimum flake density: min flake density

Minimum type-"m" density: min m-density

Minimum type-"p" density: min p-density

Maximum level: max level

Maximum E-W for "L": max E-W for L

Minimum E-W for "L": min E-W for L

Maximum E-W for "R": max E-W for R

Minimum E-W for "R": min E-W for R

Maximum N-S: max N-S

Minimum N-S: min N-S

SRE 130:

The progress information is:

File ID: file ID

Next transaction: next trans#

SRE 131:

A batch update was requested at time.

SRE 132:

The SPECS session ended at time.

D.2 TERMINAL REPORTS

Terminal Report 1:

```
The new passwords are:

     Master:  master

     Input:  input

     Change:  change

     Delete:  delete

     Cancel:  cancel

     Validation-control:  validation

     Show-acc:  show-acc

     Set-acc:  set-acc

     Update:  update
```

Terminal Report 2:

(identical to SRE 129, page 215)

Terminal Report 3:

(identical to SRE 130, page 216)

Terminal Report 4

```
         111111111122222222223333333333444444444455555555556666666666777777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

SPECIMEN       UNIT  GRID           LOCATION        COUNT  WEIGHT  DESCRIPTION
```

(The column numbers are printed here to show how the input headers fit into the standard 80-character line on a terminal, but are not part of the report.)

Terminal Report 5:

The current information about this specimen follows:

The new specimen number is _spec#_, which is encoded:

Accession number: _acc#_

Sequence and continuation number: _seq#_

Specimen type: _type_

The new site information is _site_.

The new year information is: _year_.

The new unit information is _unit info_, which is encoded:

Unit type: _unit type_

Unit: _unit_

[_added unit_ was added to the "location" field.]

The new grid square information is _grid info_, which is encoded:

Grid-square type: _gs type_

N - S coord: _N-S coord_

L - R half: _L-R half_

E - W coord: _E-W coord_

[_Grid info_ was added to the "location" field.]

The new depth information is <u>depth info</u>, which is encoded:

Depth type: <u>depth type</u>

Depth: <u>depth</u>

[<u>Depth info</u> was added to the "location" field.]

The new location information is <u>loc</u>.

The location length is <u>loc length</u>.

The new count information is <u>count info</u>, which is encoded:

Count type: <u>count type</u>

Count: <u>number</u>

The new weight information is <u>weight info</u>, which is encoded:

Weight type: <u>weight type</u>

Weight: <u>ounces</u>

The new description information is <u>descr</u>.

The description length is <u>descr length</u>.

(In this report, the lines enclosed in square brackets are to be printed only for information that will be added to the "location" field.)

D.3 UPDATE-REPORT ENTRIES

URE 1:

UPDATE REPORT

SPECS Version 1

date, time

URE 2:

SPECS could not open the system parameters file to read from it.

URE 3:

SPECS could not read the progress record from the system parameters file.

URE 4:

SPECS could not read the validation-constants record from the system parameters file.

URE 5:

SPECS could not open the system parameters file to write in it.

URE 6:

SPECS could not open the site-yr-acc file.

URE 7:

SPECS could not open the old master file.

URE 8:

SPECS could not read the old-master-file header record.

URE 9:

The "old master file" that was provided was not a master file.

URE 10:

The old master file was not the master file that should be updated.

URE 11:

SPECS could not open the transaction file.

URE 12:

SPECS could not read the transaction-file header record.

URE 13:

The "transaction file" that was provided was not a transaction file.

URE 14:

The transaction file was not the one that should be used to update the master file.

URE 15:

SPECS could not open the new master file for writing.

URE 16:

SPECS could not write the header record for the new master file.

URE 17:

SPECS could not open the new transaction file for writing.

URE 18:

SPECS could not write the header record for the new transaction file.

URE 19:

SPECS could not sort the transaction file.

URE 20:

The master file to be updated is: file ID.

The current transaction has identifications: ID list.

The new master file is: new-file ID.

The new transaction file is: new-file ID.

URE 21: The update ended at time.

No master-file or transaction-file records were processed, and no new master or transaction file was produced.

URE 22:

SPECS IS BEGINNING PROCESSING FOR SPECIMEN spec#.

The transactions to be applied are:

URE 23:

Insert-data transaction current trans#, which has these values:

Accession number: acc#
Sequence and continuation number: seg#
Specimen type: type
Transaction number: trans#
Transaction type code: I
Site: site
Year: year
Unit type: unit type
Grid-square type: gs type
Depth type: depth type
Count type: count type
Weight type: weight type
Location length: loc length
Description length: descr length
Unit: unit ID
N-S coord: N-S coord
L-R half: L-R half
E-W coord: E-W coord
Depth: depth
Count: number
Weight: ounces
Location: loc
Description: descr

(In this entry, values of fields that do not exist in the record are printed as NONE.)

URE 24:

Change-data transaction current trans#, which has these values:

Accession number: acc#

Sequence and continuation number: seq#

Specimen type: type

Transaction number: trans#

Transaction type code: C

Transaction options: options

Input field: field

Field count: field count

[Description: description]

(In this entry, [Description: description] appears only if "transaction options" is greater than 0)

URE 25:

Field designation:

field designation (field changed)

Field size: field size

Field format: field format

New value: new value

(In this entry, values of fields that do not exist in the record are printed as NONE.)

URE 26:

Delete-data transaction current trans#, which has these values:

Accession number: acc#
Sequence and continuation number: seq#
Specimen type: type
Transaction number: trans#
Transaction type code: D
Transaction options: options
[Description: description]

(In this entry, [Description: description] appears only if description information was given.)

URE 27:

Cancel-transaction record current trans#, which has these values:

Accession number: acc#
Sequence and continuation number: seq#
Specimen type: type
Transaction number: trans#
Transaction type code: X
Affected transaction number: cancel#

URE 28:

Validation-restoration transaction current trans#, which has these values:

Accession number: acc#
Sequence and continuation number: seq#

Specimen type: <u>type</u>

Transaction number: <u>trans#</u>

Transaction type code: R

Affected transaction number: <u>check#</u>

URE 29:

Validation-override transaction <u>current</u> <u>trans#</u>, which has these values:

Accession number: <u>acc#</u>

Sequence and continuation number: <u>seq#</u>

Specimen type: <u>type</u>

Transaction number: <u>trans#</u>

Transaction type code: 0

Affected transaction number: <u>confirm#</u>

URE 30:

The transaction will be ignored. It could be applied to more than one specimen-data record, but contains no information about which records it should affect.

URE 31:

The transaction was superceded by # <u>supercessor</u>, and has no effect.

URE 32:

The transaction was ignored because the transaction it affected did not exist for this specimen number.

URE 33:

The transaction was ignored. Validation can be overridden or restored only for insert-data and change-data transactions.

URE 34:

The transaction was cancelled by # <u>canceller</u>, and has no effect.

URE 35:

Validation for this transaction was overridden by transaction # <u>override</u>.

URE 36:

Validation for this transaction was restored by transaction # <u>restorer</u>.

URE 37:

The following specimen-data record, from the old master file, is being processed:

URE 38:

The following specimen-data record, created from transaction # <u>source</u>, is being processed:

URE 39:

Accession number: <u>acc#</u>
Sequence and continuation number: <u>seq#</u>
Specimen type: <u>type</u>
Site: <u>site</u>

Year: year

Unit type: unit type

Grid-square type: gs type

Depth type: depth type

Count type: count type

Weight type: weight type

Location length: loc length

Description length: descr length

Unit: unit ID

N-S coord: N-S coord

L-R half: L-R half

E-W coord: E-W coord

Depth: depth

Count: number

Weight: ounces

Location: loc

Description: descr

URE 40:

Adding to the "location" field made it longer than 127 characters. The value of the field was truncated.

URE 41:

Adding to the "description" field made it longer than 127 characters. The value of the field was truncated.

URE 42:

Transactions numbers trans list were applied to this specimen-data record.

URE 43:

Most specimens whose descriptions contain keyword are of type type, but this one is not. Changes made to its type and description are presumed to be erroneous.

URE 44:

The kind weight is greater than max weight, the maximum allowed for it by the validation constants. Changes made to the specimen's type, description, count, and weight are presumed to be erroneous.

URE 45:

The number of pieces in the specimen is greater than max count, the largest number allowed by the validation constants for kind. Changes made to the specimen's type, description, count, and weight are presumed to be erroneous.

URE 46:

The number of pieces per ounce in the specimen is not between min density and max density, the range allowed by the validation constants for kind. Changes made to the specimen's type, description, count, and weight are presumed to be erroneous.

URE 47:

No accession numbers are recorded for site(year). Changes made to the specimen number, site, and year are presumed to be erroneous.

URE 48:

Accession number acc# list is already recorded for site(year), but acc# is not. Changes made to the specimen number, site, and year are presumed to be erroneous.

URE 49:

The specimens from this location are not recorded in order of increasing depth. Changes made to the specimen number, site, year, grid square, unit, and depth information are presumed to be erroneous.

URE 50:

The value for field changed appears erroneous. It was accepted anyway because it was set by transaction # confirm#, whose validation was overridden.

URE 51:

The value for field changed reverts to old value because the new value for the field appears erroneous.

URE 52:

The unit information in the "location" field was not changed, and may now be incorrect.

URE 53:

The depth information in the "location" field was not changed, and may now be incorrect.

URE 54:

The grid-square information in the "location" field was not changed, and may now be incorrect.

URE 55:

The fields had these values after processing:

URE 56:

The record was written on the new master file.

URE 57:

Insert-data transaction trans# was written on the new transaction file. It will add the specimen to the master file during the next batch update.

URE 58:

The record was deleted.

URE 59:

The record was not written on the new transaction file because it had erroneous values.

URE 60:

Change-data or delete-data transactions trans list were not applied to any specimen-data record, and were ignored.

URE 61:

SPECS HAS FINISHED PROCESSING FOR SPECIMEN NUMBER spec#.

URE 62:

Processing for the update has ended.

add# new records were added.

delete# records were deleted.

change# records were changed, including change-spec# that will be added during the next batch update.

reject# new records were rejected because they failed validation.

There are now NMF count specimen-data records in the new master file.

URE 63:

The new master and transaction files, with identifications new-file ID, are now the current versions.

URE 64:

THE SYSTEM PARAMETERS FILE COULD NOT BE UPDATED!!!!!

URE 65:

SYSTEM PARAMETERS REPORT

URE 66:

(Identical to SRE 129, page 215)

URE 67:

(Identical to SRE 130, page 216)

D.4 INFORMATION TO BE SUBSTITUTED INTO REPORTS

a-count: the current value of the "type-"a" count" validation constant.

acc#: an accession number.

acc# list: the list of accession numbers for a given site and year in the site-yr-acc file.

added unit: the unit information added to the "location" field.

add# the number of specimen-data records added during a batch update.

b-count: the current value of the "type-"b" count" validation constant.

bead count: the current value of the "bead count" validation constant.

cancel: the current "cancel" password.

cancel#: the number of a transaction to be cancelled.

canceller: the transaction number of a cancel-transaction record.

change: the current "change" password.

change#: the number of specimen-data records changed during a batch update.

change-spec#: the number of specimen-data records whose specimen numbers were changed during a batch update.

check#: the number of a transaction for which a CHECK command was given.

command: a SPECS command.

comment: the text of a comment, provided by the user as part of a COMMENT command.

confirm#: the number of a transaction for which a CONFIRM command was given.

constant: the name of a validation constant, as given in Appendix B.

constant value: the value of a validation constant.

container count: the current value of the "container count" validation constant.

count: a count of the number of things in a specimen, after single-field validation.

count info: information given by the user as a value for the "count" input field.

count type: the current value of the "count type" field in a master-file or transaction-file record.

current trans#: the number of the transaction being processed when an update-report entry is written.

date: the date and day of the week of the interactive session or batch update by which the session or update report was produced.

delete: the current "delete" password.

delete#: the number of specimen-data records deleted during a batch update.

depth: the current value of the "depth" field in a master-file or transaction-file record.

depth info: information given by the user as a value for the "depth" input field.

depth type: the current value of the "depth type" field in a master-file or transaction-file record.

descr: the current value of the "description" field in a master-file or transaction-file record.

description: a part of a specimen's description, given by the user to aid in recognizing the specimen during a batch update.

description info: information given by the user as a value for the "description" input field.

descr length: the current value of the "description length" field in a master-file or transaction-file record.

e-count: the current value of the "type-"e" count" validation constant.

E-W coord: the current value of the "E-W coord" field in a master-file or transaction-file record.

field: the name of an input field.

field changed: the name, as given in Table 5 (Appendix B), of a specimen-data field whose value is being changed.

field count: the current value of the "field count" field in a change-data record.

field designation: the current value of a "field designation" field in a change-data record.

field format: the current value of a "field format" field in a change-data record.

field size: the current value of a "field size" field in a change-data record.

<u>file date</u>: the date on which the current transaction file was created.

<u>file ID</u>: the identification of the current master and transaction files; also, the current value of "file ID" in the progress record of the system parameters file.

<u>flake count</u>: the current value of the "flake count" validation constant.

<u>grid info</u>: information given by the user as a value for the "grid" input field.

<u>gs type</u>: the current value of the "grid-square type" field in a master-file or transaction-file record.

<u>h-count</u>: the current value of the "type-"h" count" validation constant.

<u>ID list</u>: the list of the transaction-file identifications in a transaction file.

<u>input</u>: the current "input" password.

<u>keyword</u>: a word or phrase from Table 10 (Appendix C) that, when used in a specimen's description, usually indicates that the specimen is of a particular type.

kind: a specimen kind, used in kind-count-weight validation.

L-R half: the current value of the "L-R half" field in a master-file or transaction-file record.

loc: the current value of the "location" field in a master-file or transaction-file record.

loc length: the current value of the "location length" field in a master-file or transaction-file record.

location info: information given by the user as a value for the "location" input field.

m-count: the current value of the "type-"m" count" validation constant.

master: the current "master" password.

max a-density: the current value of the "maximum type-"a" density" validation constant.

max b-density: the current value of the "maximum type-"b" density" validation constant.

max bead density: the current value of the "maximum bead density" validation constant.

<u>max</u> <u>count</u>: the current value of a validation constant that represents the maximum number of pieces in a specimen of a given kind.

<u>max</u> <u>density</u>: the current value of a validation constant that represents the maximum number of pieces per tenth of an ounce in a specimen of a given kind.

<u>max</u> <u>e-density</u>: the current value of the "maximum type-"e" density" validation constant.

<u>max</u> <u>E-W</u> <u>for</u> <u>L</u>: the current value of the "maximum E-W for "L"" validation constant.

<u>max</u> <u>E-W</u> <u>for</u> <u>R</u>: the current value of the "maximum E-W for "R"" validation constant.

<u>max</u> <u>flake</u> <u>density</u>: the current value of the "maximum flake density" validation constant.

<u>max</u> <u>h-density</u>: the current value of the "maximum type-"h" density" validation constant.

<u>max</u> <u>level</u>: the current value of the "maximum level" validation constant.

<u>max</u> <u>m-density</u>: the current value of the "maximum type-"m" density" validation constant.

<u>max</u> <u>N-S</u>: the current value of the "maximum N-S" validation constant.

max p-density: the current value of the "maximum type-"p" density" validation constant.

max weight: the current value of the "soil sample weight" or "washings weight" validation constant.

min a-density: the current value of the "minimum type-"a" density" validation constant.

min b-density: the current value of the "minimum type-"b" density" validation constant.

min bead density: the current value of the "minimum bead density" validation constant.

min density: the current value of a validation constant that represents the minimum number of pieces per tenth of an ounce in a specimen of a particular kind.

min e-density: the current value of the "minimum type-"e" density" validation constant.

min E-W for L: the current value of the "minimum E-W for "L"" validation constant.

min E-W for R: the current value of the "minimum E-W for "R"" validation constant.

min flake density: the current value of the "minimum flake density" validation constant.

min h-density: the current value of the "minimum type-"h" density" validation constant.

min m-density: the current value of the "minimum type-"m" density" validation constant.

min N-S: the current value of the "minimum N-S" validation constant.

min p-density: the current value of the "minimum type-"p" density" validation constant.

N-S coord: the current value of the "N-S coord" field in a master-file or transaction-file record.

new-file ID: the identification of the new master and transaction files produced by a batch update.

new value: the current value of a "new value" field in a change-data record.

next trans#: the next transaction number to be assigned.

NMF-count: the number of specimen-data records written on the new master file during a batch update.

number: the current value of the "count" field in a master-file or transaction-file record.

old CSPP: a variant spelling of "CSPP" given by the user in information for the "description" input field.

old spec#: the specimen number of the "previous specimen."

old value: the value of a field at the beginning of a batch update.

options: the current value of the "transaction options" field of a change-data or delete-data record.

ounces: the current value of the "weight" field in a master-file or transaction-file record.

override: the transaction number of a validation-override record.

p-count: the current value of the "type-"p" count" validation constant.

parameters: the current "parameters" password.

password: the name of a password.

record: the name of a record in the system parameters file.

reject#: the number of records that were not added to the master file because they failed batch-update validation.

reset name: the name of a password or validation constant whose value is to be reset.

restorer: the transaction number of a validation-restoration record.

seq#: a sequence and continuation number.

set-acc: the current "set-acc" password.

show-acc: the current "show-acc" password.

site: a site number.

soil weight: the current value of the "soil sample weight" validation constant.

source: the transaction number of the insert-data record used to create a new specimen-data record during a batch update.

spec#: a specimen number.

supercessor: the number of a superceding transaction in a batch update.

time: the time of day at which a session or update report entry was generated.

trans list: a list of transaction numbers.

trans#: the transaction number of the record most recently written on the current (for session reports) or new (for update reports) transaction file.

type: a specimen type.

unit#: the numeric part of a unit number.

unit ID: the current value of the "unit" field in a master-file or transaction-file record.

unit info: information given by the user as a value for the "unit" input field.

unit type: the current value of the "unit type" field in a master-file or transaction-file record.

update: the current "update" password.

username: the user's name.

validation: the current "validation-control" password.

value: information given by the user in response to a prompt.

washings weight: the current value of the "washings weight" validation constant.

<u>weight info</u>: information given by the user as a value for the "weight" input field.

<u>weight type</u>: the current value of the "weight type" field in a master-file or transaction-file record.

<u>year</u>: a year.

Appendix E

INTERACTIVE-SESSION PSEUDOCODE

E.1 OVERVIEW

This appendix is a detailed description of the output produced by an interactive session. It gives SPECS' side of the dialogue with the user, and the resulting session-report entries, terminal reports, and records written in files.

As part of the specification, the appendix includes an algorithm that dictates what output is produced when. The implementor may modify the algorithm in any way he wishes, as long as he does not affect the relationship of output to input. In particular, he should use subroutines extensively.

E.2 NOTATION

The pseudocode in the appendix is written as a hierarchy of steps, each intended to perform an action. High-level steps are too complex to be readily comprehended by reading their descriptions. Lower-level steps divide the high-level steps into sequences of simpler activities. The lower-level steps may themselves be divided until, at the lowest level, the actions to be performed are simple enough to need no further explication.

Each step has a unique number that identifies both the higher-level step of which it is a part and its position in the sequence of subdivisions of the higher-level step. The number for each subdivision of a step concatenates the step number to a period, then to the sequence number for the subdivision. Step numbers for the highest-level steps are integers.

The pseudocode for step 1 (pages 253 to 273) illustrates the notational conventions used in the rest of this appendix. Step 1 is divided into four substeps (1.1, 1.2, 1.3, and 1.4). Steps 1.2, 1.3, and 1.4 are further subdivided.

Messages that are to be printed at the terminal are enclosed in quotation marks. Examples are found in steps 1.1 and 1.2.3.

Session-report entries are written by steps whose pseudocode includes "SRE" followed by the number of a session-report entry listed in Appendix D. Step 1.2.2 is an example of a step that writes a session-report entry.

Step 1.3.3 is a typical "select" step. Executing it requires selecting the first true condition from among a series of conditions in which at least one is always true, then performing the action specified by the pseudocode for the true condition. If the user's response to step 1.3.2 was a colon, for example, executing step 1.3.3 prints the menu listed in the pseudocode for the colon, then restarts step 1.3.1.

Steps whose pseudocode includes "Terminal Report" print one of the terminal reports in Appendix D at the terminal. These reports are like the messages in quotation marks, but are too long to be conveniently listed in the pseudocode. (Step 1.4.4.4.1 prints a terminal report.)

E.3 CONTENTS

Table 12 lists the major steps executed during interactive sessions, and where the pseudocode for them may be found.

TABLE 12

Activities and steps in interactive pseudocode

Activity	Steps	Page
Initialization	1 - 4	253
All sessions	1	253
System-parameters file reconstruction	1.4.4	257
Master-file reconstruction	3	276
Command interpretation	4	282
Command interpretation	5	285
Getting passwords	5.3	285
Menu	5.4	288
Explanation	5.5	289
CANCEL command	5.6	290
CHANGE command	5.7	295
Decoding the command	5.7.1 - 5.7.6	296
Getting the new value	5.7.7	305
Validation	5.7.9	307
Writing the change-data record	5.7.10 - 5.7.12	336
Showing the new values	5.7.14	363
CHECK command	5.8	371
COMMENT command	5.9	377
CONFIRM command	5.10	378
DELETE command	5.11	384
END command	5.12	392

E.4 PSEUDOCODE

```
1.  Begin initialization.

    1.1  Write welcoming messages at the terminal.

                    "Welcome to Version 1 of SPECS,

                     a storage and retrieval system

                       for specimen catalog data


                The Research Laboratories of Anthropology

                                  at the

               University of North Carolina at Chapel Hill."


                      [date and day of week of session]


         "Please wait."

    1.2  Open the session  report file and write  the session
         report header on it.   If the header can't be written,
         write an  error message  at the  terminal,  then  stop
         initialization.

         1.2.1  Open the session report file for writing.
```

1.2.2 If the opening succeeds, write the session report header. (SRE 1)

1.2.3 If opening fails, write an error message at the terminal, then stop initialization.

"SPECS cannot open the session report file."

1.3 Ask the user for his name, and write it on the session report.

1.3.1 Write a prompt at the terminal.

"Please enter your name:"

1.3.2 Wait for the user's response.

1.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to end this SPECS session,

? for an explanation, or

your name."

Q, print messages at the terminal and on the session report, and stop initialization.
(SRE 2)

"At your request, SPECS is stopping this session."

?, print an explanation at the terminal and repeat the prompt.

"You are beginning a SPECS interactive session. SPECS wants to know your name, which it will print on the session report."

anything else, consider it the user's name, and write it on the session report. (SRE 3)

1.4 Read the system parameters file and open it for writing. If it cannot be read, try to reconstruct it. If reconstruction fails or the file cannot be opened for writing, write error messages and stop initialization.

1.4.1 Open the system parameters file for reading. If it can't be read, write error messages at the terminal and on the session report. (SRE 4)

"SPECS cannot open the system parameters file."

1.4.2 If the system parameters file was opened, read it.

1.4.2.1 Read the progress record. If it cannot be read, write error messages at the terminal and on the session report, and stop trying to read the system parameters file. (SRE 5)

"SPECS cannot read the progress information from the system parameters file."

1.4.2.2 Read the validation-constants record. If it cannot be read, write error messages at the terminal and on the session report, and stop trying to read the system parameters file. (SRE 6)

"SPECS cannot read the validation constants from the system parameters file."

1.4.2.3 Read the passwords record. If it cannot be read, write error messages at the terminal and on the session report, and stop trying to read the system parameters file. (SRE 7)

"SPECS cannot read the passwords from the system parameters file."

1.4.3 Open the system parameters file for writing. If it cannot be opened, write error messages at the terminal and on the session report, and stop initialization. (SRE 8)

"SPECS cannot write in the system parameters file."

1.4.4 If the system parameters file could not be read but was opened for writing, try to reconstruct the file by asking the user for the information it contained. If the user does not supply the information, stop initialization.

1.4.4.1 Tell the user what's happening.

"SPECS will try to reconstruct the system parameters file by asking you for the missing information."

1.4.4.2 Ask the user for the parameters password, knowledge of which demonstrates his authority to supply new values for file reconstruction. If the password is not given, stop initialization.

1.4.4.2.1 Print a prompt at the terminal.

"Please enter the parameters password:"

1.4.4.2.2 Wait for the user's response.

1.4.4.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to end this SPECS session,

? for an explanation, or

the password."

?, print an explanation at the terminal and repeat the prompt.

"SPECS could not read the system parameters file, but may be able to reconstruct it with your help. If you supply the parameters password to demonstrate that you are authorized to give values for the system parameters file, SPECS will ask you for information to reconstruct the file."

Q, print messages at the terminal and on the session report, and stop initialization. (SRE 9)

"At your request, SPECS is stopping its attempt to reconstruct the system parameters file."

the correct password, remember that it was given.

anything else (the first 2 times an incorrect password is given), print a message at the terminal and repeat the prompt.

"SPECS does not recognize your response as the correct password."

anything else (the third time), print messages at the terminal and on the session report, and stop initialization. (SRE 9)

"SPECS did not receive the correct password, and will not reconstruct the system parameters file."

1.4.4.3 Ask the user for a new value for each of the SPECS passwords stored in the system parameters file. There are nine such passwords: master, input, change, delete, cancel, validation-control, show-acc, set-acc, and update. If the user refuses to supply them, stop initialization.

In the following, <u>name</u> is replaced by the password's name, and <u>function</u> is replaced by a description of its function, from Table 13.

TABLE 13

Names and functions of passwords

Name	Function
master	reset passwords and progress information
input	give SPECS information about new specimens
change	change information that is already entered
delete	delete all information about any specimen
cancel	cancel any request that has been made but not yet filled
validation-control	tell SPECS to ignore or abide by the results of validation during a batch update
show-acc	tell SPECS to show the accession number for a site and year
set-acc	tell SPECS to enter or change data in the site-yr-acc file
update	tell SPECS to update the master file

1.4.4.3.1 Print a prompt for the password at the terminal.

"Please enter the <u>name</u> password:"

1.4.4.3.2 Wait for the user's response.

1.4.4.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to end this SPECS session,

? for an explanation,

: to repeat this menu, or

a value for the password."

?, print an explanation at the terminal and repeat the prompt.

"SPECS could not read the system parameters file, and is getting the missing values from you. It wants you to tell it the <u>name</u> password, which controls who can <u>function</u>."

Q, print messages at the terminal and on the session report, and stop initialization. (SRE 10)

"At your request, SPECS is stopping its attempt to reconstruct the system parameters file."

any other series of 1-8 characters, consider it the password.

anything else, print a message at the terminal and repeat the prompt.

"All passwords must have 1 - 8 characters."

1.4.4.4 Write a report at the terminal to summarize the new values of the passwords, and tell the user how to change them.

1.4.4.4.1 Print a new-passwords report at the terminal. (Terminal Report 1).

1.4.4.4.2 Print a message at the terminal to tell the user how to change these values.

"These values may be changed by using the RESET command and the value shown above for the master password."

1.4.4.5 Ask the user for a new value for each of the validation constants. If the user refuses to supply the values, stop initialization.

In the following steps, name and function are replaced by the name and function of the constant, as listed in Table 14.

TABLE 14

Names and functions of validation constants

Name	Function
soil-sample weight	the maximum weight for a soil sample
washings weight	the maximum weight for washings
bead count	the maximum number of beads in a specimen
type-"a" count	the maximum number of pieces other than beads in a specimen of type "a"
type-"b" count	the maximum number of pieces in a specimen of type "b"
type-"e" count	the maximum number of pieces in a specimen of type "e"
type-"h" count	the maximum number of pieces in a specimen of type "h"
flake count	the maximum number of flakes and chips in a specimen of type "m"
type-"m" count	the maximum number of pieces other than flakes, chips, soil samples, and washings in a specimen of type "m"
type-"p" count	the maximum number of pieces in a specimen of type "p"
container count	the maximum value of "count" for specimens counted in boxes, jars, bags, or vials
maximum bead density	the maximum number of beads per tenth of an ounce
maximum type-"a" density	the maximum number of pieces other than beads per tenth of an ounce in specimens of type "a"

maximum type-"b" density	the maximum number of pieces per tenth of an ounce in specimens of type "b"
maximum type-"e" density	the maximum number of pieces per tenth of an ounce in specimens of type "e"
maximum type-"h" density	the maximum number of pieces per tenth of an ounce in specimens of type "h"
maximum flake density	the maximum number of flakes and chips per tenth of an ounce in specimens of type "m"
maximum type-"m" density	the maximum number of pieces other than flakes and chips per tenth of an ounce in specimens of type "m" other than soil samples and washings
maximum type-"p" density	the maximum number of pieces per tenth of an ounce in specimens of type "p"
minimum bead density	the minimum number of beads per tenth of an ounce
minimum type-"a" density	the minimum number of pieces other than beads per tenth of an ounce in specimens of type "a"
minimum type-"b" density	the minimum number of pieces per tenth of an ounce in specimens of type "b"
minimum type-"e" density	the minimum number of pieces per tenth of an ounce in specimens of type "e"
minimum type-"h" density	the minimum number of pieces per tenth of an ounce in specimens of type "h"
minimum flake density	the minimum number of flakes and chips per tenth of an ounce in specimens of type "m"

minimum type-"m" density	the minimum number of pieces other than flakes and chips per tenth of an ounce in specimens of type "m" other than soil samples and washings
minimum type-"p" density	the minimum number of pieces per tenth of an ounce in specimens of type "p"
maximum level	the maximum level or zone number
maximum E-W for "L"	the maximum east-west coordinate in the left half of a site
minimum E-W for "L"	the minimum east-west coordinate in the left half of a site
maximum E-W for "R"	the maximum east-west coordinate in the right half of a site
minimum E-W for "R"	the minimum east-west coordinate in the right half of a site
maximum N-S	the maximum north-south coordinate in a site
minimum N-S	the minimum north-south coordinate in a site

1.4.4.5.1 Write a prompt for the validation constant at the terminal.

"Please enter the value for <u>name</u>:"

1.4.4.5.2 Wait for the user's response.

1.4.4.5.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to end this SPECS session, or

the <u>name</u>."

?, print an explanation at the terminal and repeat the prompt.

"SPECS could not read the system parameters file, and is getting the missing values from you. It wants a value for <u>function</u>. The value will be used during validation."

Q, print messages at the terminal and on the session report, and stop initialization. (SRE 11)

"At your request, SPECS is stopping its attempt to reconstruct the system parameters file."

a value with the characteristics listed in Appendix B for the validation constant, consider it the new value for the constant.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your answer."

1.4.4.6 Write a report at the terminal to summarize the new values of the validation constants, and tell the user how to change them.

1.4.4.6.1 Print a validation-constants report at the terminal. (Terminal Report 2)

1.4.4.6.2 Print a message at the terminal to tell the user how to change these values.

"These values may be changed by using the RESET command and the value you gave for the validation-control password."

1.4.4.7 Ask the user for the new progress information. If he refuses to supply it, stop initialization.

1.4.4.7.1 Ask the user for the identification of the current master and transaction files.

1.4.4.7.1.1 Write a prompt at the terminal.

"Please enter the identification number of the current master and transaction files, as shown on the most recent session or update report:"

1.4.4.7.1.2 Wait for the user's response.

1.4.4.7.1.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to end this SPECS session, or

the identification number of the master and transaction files."

?, print an explanation at the terminal and repeat the prompt.

"SPECS could not read the system parameters file, and is getting the missing values from you. It wants to know the identification number of the master and transaction files to be used in the next batch update."

Q, print messages at the terminal and on the session report, and stop initialization. (SRE 12)

"At your request, SPECS is stopping its attempt to reconstruct the system parameters file.

a non-negative integer with one to ten digits, assume it is the file identification.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your answer."

1.4.4.7.2 Ask the user for the next transaction number to be assigned.

1.4.4.7.2.1 Write a prompt at the terminal.

"Please enter the next transaction number to be assigned, as it is shown on the most recent session or update report:"

1.4.4.7.2.2 Wait for the user's response.

1.4.4.7.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to end this SPECS session, or

the next transaction number to be assigned."

?, print an explanation at the terminal and repeat the prompt.

"SPECS could not read the system parameters file, and is getting the missing values from you. It needs to know the next transaction number to be assigned, so that it can avoid assigning duplicate numbers."

Q, print messages at the terminal and on the session report, and stop initialization. (SRE 13)

"At your request, SPECS is stopping its attempt to reconstruct the system parameters file."

a non-negative integer greater than the file identification, assume it is the next transaction number.

a non-negative integer less than or equal to the file identification, print a message at the terminal and repeat the prompt.

"The next transaction number must be higher than the identification of the master and transaction files."

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your answer."

1.4.4.8 Write a report at the terminal to summarize the progress information. (Terminal Report 3)

1.4.4.9 Give the user an opportunity to verify the progress information. If he refuses, get new progress information.

1.4.4.9.1 Print a question at the terminal.

"Please check these values very carefully. Are they correct?"

1.4.4.9.2 Wait for the user's response.

1.4.4.9.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter YES to confirm that all of the
values are correct,
NO if at least one is wrong,
Q to end the SPECS session, or
? for an explanation."

?, print an explanation at the terminal and repeat the question.

"SPECS could not read the system parameters file, and is getting the missing values from you. It needs to be certain that the progress information it has just repeated is correct."

YES, do nothing.

NO, get the progress information again by restarting step 1.4.4.7.

Q, print messages at the terminal and on the session report, and stop initialization. (SRE 14)

"At your request, SPECS is stopping its attempt to reconstruct the system parameters file."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

1.4.5 If the system parameters file was reconstructed, write the new values into the file.

1.4.5.1 Write the progress record.

1.4.5.2 Write the validation-constants record.

1.4.5.3 Write the passwords record.

1.4.5.4 If any of these records could not be written, write messages at the terminal and on the session report. (SRE 15)

"SPECS cannot write the new values in the system parameters file."

1.4.5.5 If the file was reconstructed and written, write messages at the terminal and on the session report. (SRE 16)

"System parameters file reconstruction is complete."

2. If initialization has not been stopped, ask the user whether this is a normal SPECS session (intended to generate transaction-file records), or is intended to reset values used during master-file reconstruction.

   2.1 Print a question at the terminal.

   "Is this session to set the values used in master-file reconstruction?"

   2.2 Wait for the user's response.

   2.3 If the response was:

   :, print a menu at the terminal and repeat the question.

   "Enter YES if you need to reconstruct a master file
   or reset the information you changed for
   the reconstruction,

   NO if you do not,

   Q to end this SPECS session, or

   ? for an explanation."

   ?, print an explanation at the terminal and repeat the question.

   "SPECS wants to know whether this is a normal session (one in which you will change or enter data, reset system parameters or validation constants, or request a batch update), or one intended to prepare SPECS to reconstruct a lost master file or reset the file identification and next transaction number after a reconstruction. If you specify that you are setting the values needed for master-file reconstruction, SPECS will let you reset only information needed to control its behavior during reconstruction. Otherwise, SPECS will expect to add to the current transaction file, and will not allow you to proceed unless you have provided that file."

YES, remember that the session is for reconstruction, and print a session report entry to say so. (SRE 17)

NO, remember that the session is normal.

Q, print messages at the terminal and on the session report, and stop initialization. (SRE 18)

"At your request, the SPECS session is ending."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

3. If that this session is for master-file reconstruction, ask the user for the new file identification and next transaction number.

  3.1 Ask the user for the master password. If he does not supply it, print a message on the session report and do not allow him to change the file identifications.

    3.1.1 Print a prompt at the terminal.

      "Please enter the master password:"

    3.1.2 Wait for the user's response.

    3.1.3 If the response was:

      :, print a menu at the terminal and repeat the prompt.

        "Enter Q to end this SPECS session,

        ? for an explanation, or

        the password."

      ?, print an explanation at the terminal and repeat the prompt.

        "You have told SPECS that this session is to give it information that it needs to reconstruct a master file or to reset information that was changed for the reconstruction. SPECS wants you to tell it the master password to demonstrate that you are authorized to give it the information it needs."

Q, print messages at the terminal and on the session report, and stop trying to get the master password. (SRE 19)

"At your request, SPECS is stopping its attempt to get information for master-file reconstruction."

the correct password, remember that the user knows it.

anything else (the first 2 times an incorrect password is given), print a message at the terminal and repeat the prompt.

"SPECS cannot recognize your response as the correct password."

anything else (the third time), print messages at the terminal and on the session report, and stop trying to get the master password. (SRE 19)

"SPECS did not receive the correct password, and cannot accept information for master-file reconstruction."

3.2 Print a message at the terminal to tell the user about the current file identification.

In the message, <u>file</u> <u>ID</u> is the file identification from the progress record of the system parameters file.

"The current file identification is: <u>file</u> <u>ID</u>.

3.3 Ask the user for the identification of the master and transaction files to be used during the next batch update. If he refuses to identify the files, assume that the file identifications and next transaction number should remain unchanged.

3.3.1 Print a prompt at the terminal.

"Please enter the identification number of the master and transaction files that should be used in the next batch update:"

3.3.2 Wait for the user's response.

3.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

SAME if the file identification should remain unchanged,

Q to end this SPECS session, or

a new identification number for the master and transaction files."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS that this session is to give it the information it needs to reconstruct a master file or to reset information that was changed for the reconstruction. SPECS will use the master and transaction files whose identification you provide now for its next batch update."

SAME, do nothing.

Q, print messages at the terminal and on the session report, and stop getting information for master-file reconstruction. (SRE 20)

"At your request, SPECS is stopping its attempt to reconstruct the master file or reset information afterwards."

a non-negative integer with 1 to 10 digits, assume it is the file identification.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your answer."

3.4 Ask the user for the next transaction number to be assigned. If this session is in preparation for file reconstruction, the next transaction number is the identification of the new master and transaction files to be produced by the reconstruction. If the user refuses to identify the files, assume that the identification of the old and new master and transaction files should remain unchanged.

3.4.1 Print a prompt at the terminal.

"Please enter the identification number of the new master and transaction files to be produced by the next batch update:"

3.4.2 Wait for the user's response.

3.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to end this SPECS session, or

the identification number of the master and transaction files."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS that this session is to give it the information it needs to reconstruct a master file or to reset the information that was changed for the reconstruction. SPECS now wants you to tell it the next transaction number to be assigned. If you are reconstructing a master file, this number is the file identification of the new master file to be produced by the batch update. If you are resetting information after the reconstruction, the next transaction number is one higher than the number of the last transaction that was written during an interactive session."

Q, print messages at the terminal and on the session report, and stop getting information for master-file reconstruction. (SEE 21)

"At your request, SPECS is stopping its attempt to reconstruct the master file or reset information afterwards."

a non-negative integer not greater than the file identification, print a message at the terminal and get the file identification again by restarting step 3.2.

"The next transaction number to be assigned must always be higher than the identification of the files for the next batch update."

a non-negative integer of not more than 10 digits that is greater than the file identification, assume it is the next transaction number.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your answer."

3.5 Write a session report entry about the file identification and next transaction number. (SRE 22)

4. If this is a normal session, finish initialization.

4.1 Open the site-year-acc file. If it cannot be opened, print messages at the terminal and on the session report, and stop initialization. (SRE 23)

"SPECS cannot open the site-year-acc file."

4.2 Prepare the transaction file for new transactions. If it cannot be prepared, tell the user and stop initialization.

4.2.1 Open the transaction file for reading. If it cannot be opened, print messages at the terminal and on the session report, and stop initialization. (SRE 24)

"SPECS cannot open the current transaction file."

4.2.2 Read the first record on the transaction file, assuming that it is formatted as a transaction-file header. If there is no such record, write messages at the terminal and on the session report, and stop initialization. (SRE 25)

"SPECS cannot read the transaction file."

4.2.3 Make sure that the record is a transaction-file header by confirming that its specimen type is "T". If it isn't, write messages at the terminal and on the session report, and stop initialization. (SRE 26)

"The file provided for transactions is not a transaction file."

4.2.4 Make sure that the transaction file is current by comparing its Xaction-file ID with the file ID from the progress record of the system parameters file. If they are not the same, write messages at the terminal and on the session report, and stop initialization. (SRE 27)

"The transaction file is not current."

4.2.5 Find the end of the file, reading at least the specimen type of each record. If records with specimen types of "T" are found, they are transaction-file headers. Remember the Xaction-file IDs of these records, write them in messages at the terminal and on the session report, and stop initialization. (SRE 28)

In the following message to be written at the terminal, <u>ID</u> <u>list</u> is replaced by a list of the Xaction-file IDs from the header records.

"The transaction file is a multiple-update file, and cannot be used for new transactions. It contains transaction files with identifications <u>ID list</u>."

4.2.6 Open the transaction file for writing, without rewinding it. If it cannot be opened, write messages at the terminal and on the session report, and stop initialization. (SRE 29)

"Although SPECS found the correct transaction file, it cannot open the file for writing."

4.2.7 Write messages at the terminal and on the session report to identify the transaction file used for the session. (SRE 30)

In the following message to be written at the terminal, <u>file</u> <u>ID</u> is replaced by the Xaction-file ID from the transaction-file header record, and <u>date</u> is replaced by the file creation date.

"The current transaction file is <u>file</u> <u>ID</u>, begun <u>date</u>."

5. If initialization has not been stopped and this is a normal session, interpret commands. Repeatedly ask for commands and execute them, then stop when an END command is received.

5.1 Print a request for a command at the terminal.

"Please enter a command:"

5.2 Wait for the user's response.

5.3 If the first word of the response was CANCEL, CHANGE, CHECK, CONFIRM, DELETE, INPUT, SET-ACC, or SHOW-ACC, confirm that the user knows the password needed to give the command. (RESET also requires a password but, because SPECS cannot tell which password is needed until it has more information about what will be reset, getting a password for RESET is postponed until the information is available.)

In the following messages to be printed at the terminal, command is replaced by the first word of the response, and password by the name of the corresponding password, from Table 15.

TABLE 15

Commands and passwords

Command	Password
CANCEL	cancel
CHANGE	change
CHECK or CONFIRM	validation-control
DELETE	delete
INPUT	input
SET-ACC	set-acc
SHOW-ACC	show-acc

If the correct password was not given earlier in the session, ask the user for it.

5.3.1 Print a prompt at the terminal.

"Please enter the password password:"

5.3.2 Wait for the user's response.

5.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to request that SPECS ignore the
<u>command</u> command,

? for an explanation, or

the correct password."

?, print an explanation at the terminal and repeat the prompt.

"You have given SPECS a <u>command</u> command. SPECS wants you to give it the current <u>password</u> password to demonstrate your authority to give the command."

Q, print messages at the terminal and on the session report, and repeat the request for a command by restarting step 5. (SRE 31)

"At your request, SPECS is ignoring the <u>command</u> command."

the correct password, remember that it has been given.

anything else (the first two times an incorrect password is given), print a message at the terminal and repeat the prompt.

"Your response is not the correct password."

anything else (the third time), print messages at the terminal and on the session report, and return to the beginning of step 5. (SRE 31)

"You have not given the correct password. SPECS is ignoring your <u>command</u> command."

5.4 If the response to the request for a command was:

:, print a menu at the terminal and repeat the request for a command.

"Enter END to end this SPECS session,

? for an explanation,

CANCEL to request that a transaction be ignored,

CHANGE to change specimen-catalog data that has already been given,

CHECK to request that a transaction be performed only if it passes validation during a batch update, COMMENT to request that a comment be written on the session report,

CONFIRM to request that a transaction be performed even if it fails validation during a batch update,

DELETE to remove all information about a specimen,

INPUT to give information about new specimens,

RESET to change passwords or validation constants,

SET-ACC to enter or change the accession number for a site and year, or

SHOW-ACC to see the accession number for a site and year."

5.5 If the response to the request for a command was:

?, print an explanation at the terminal and repeat the request for a command.

"SPECS expects you to give it a command. It will continue to request and execute commands until you use the END command to indicate that you have finished giving commands for this session."

5.6 If the response to the request for a command was:

CANCEL, write a cancel-transaction record and repeat the request for a command.

5.6.1 If there is no more information in the command, the user must be asked for the transaction and specimen numbers of the transaction to be cancelled.

5.6.2 If more information was given in the command, analyze it.

5.6.2.1 Assume the second word of the command (the one after CANCEL) is a transaction number.

5.6.2.2 If the assumed transaction number is not an integer between 0 and the next transaction number to be assigned, the user must be asked for the transaction number.

5.6.2.3 If the assumed transaction number is the last word in the command, the user must be asked for the specimen number.

5.6.2.4 If there is more information in the command, analyze it to find the specimen number.

5.6.2.4.1 If the first word in the rest of the command is FOR, assume everything following it is a specimen number.

5.6.2.4.2 If the first word is not FOR, the user must be asked for a specimen number.

5.6.2.4.3 If there is an assumed specimen number, validate it with the algorithm in Appendix C. If the validation fails, or if the specimen number has no accession number, the user must be asked for a specimen number.

5.6.3 If earlier processing shows it is needed, ask the user for the transaction number.

5.6.3.1 Print a prompt at the terminal.

"Please enter the transaction number:"

5.6.3.2 Wait for the user's response.

5.6.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CANCEL command ignored, or

the transaction number of the transaction to be cancelled."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS not to execute a transaction in the current transaction file. SPECS wants to know the number of the transaction that should not be executed."

Q, print a message at the terminal, and stop executing the CANCEL command.

"At your request, the CANCEL command is being ignored.

an integer betwen 0 and the next transaction number to be assigned, consider it the transaction number.

anything else, print a message at the terminal and repeat the prompt.

"SPECS does not recognize your response as a transaction number."

5.6.4 If earlier processing shows that it is needed, ask the user for the specimen number.

5.6.4.1 Print a prompt at the terminal.

"Please enter the specimen number:"

5.6.4.2 Wait for the user's response.

5.6.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CANCEL command ignored, or

the specimen number, including accession number, of the specimen that would be affected by the transaction to be cancelled."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to not to execute a transaction in the current transaction file. SPECS wants to know the number of the specimen that would be affected by the transaction."

Q, print a message at the terminal, and stop executing the CANCEL command.

"At your request, the CANCEL command is being ignored."

anything else, assume it is a specimen number.

5.6.4.4 Validate the assumed specimen number, using the algorithm in Appendix C. If validation fails, print a message at the terminal and repeat the request for a specimen number by restarting step 5.6.4.

"SPECS does not recognize your response as a specimen number."

5.6.4.5 If validation succeeded and the specimen number did not include an accession number, print a message at the terminal and repeat the request for a specimen number by restarting step 5.6.4.

"The specimen number must include an accession number."

5.6.5 Write a session report entry summarizing the information in the command. (SRE 32)

5.6.6 Write a cancel-transaction record on the transaction file, using these values:

Accession number, sequence and continuation number, and specimen type: from validation of the specimen number.

Transaction number: the next transaction number to be assigned.

Transaction type code: X

Affected transaction number: the number of the transaction to be cancelled.

5.6.7 Increment the next transaction number to be assigned.

5.6.8 Write messages about the cancel-transaction record at the terminal and on the session report. (SRE 33)

In the message to be written at the terminal, cancel# is replaced by the number of the transaction to be cancelled, spec# is replaced by the specimen number, and trans# is replaced by the transaction number of the cancel-transaction record.

"Transaction # cancel#, referring to specimen spec#, will be ignored during the next batch update.
(Transaction # trans#)."

5.7 If the response to the request for a command was:

CHANGE, write a change-data record and repeat the request for a command.

5.7.1 If no more information was given in the command, the user must be asked for the input field, specimen number, and description of the specimen to be changed.

5.7.2 If more information was given in the command, analyze it.

5.7.2.1 If the second word in the command (the one after CHANGE) is ALL, set the "transaction options" field of the change-data record to 0, and look at the next word of the command.

5.7.2.2 Treat the current word of the command as the name of an input field. If it is not, the command is garbled, so the user must be asked for the input field, specimen number, and description.

5.7.2.2.1 Identify the input field whose value is to be changed by finding the current word of the command in the "field" column of Table 16. If it is there, set the "input field" of the change-data record to the corresponding value from the "code" column.

TABLE 16

Input fields and codes for CHANGE commands

Field	Code
Count	1
Depth	2
Description	3
Grid	4
Location	5
Site	6
Specimen	7
Unit	8
Weight	9
Year	10

5.7.2.2.2 If the current word is not in the "field" column, it does not represent an input field. The user must be asked for the name of the input field to be changed.

5.7.2.3 Analyze the next two words of the command to find the number of the specimen to be changed.

5.7.2.3.1 The first of the two words should be "FOR". If it is not, the user must be asked for the specimen number.

5.7.2.3.2 If the first of the two words is "FOR", use the algorithm in Appendix C to validate the second of the words as a specimen number. If the validation fails, or if there is no accession number, the user must be asked for the specimen number.

5.7.2.4 If the command does not end with the specimen number, analyze the rest of the command to find a description.

5.7.2.4.1 If the first word of the rest of the command is WITH and the remainder of the command is enclosed in quotation marks, anything inside the quotation marks is a description. In this case, set the "transaction options" field of the change-data record to the length (in characters) of the description and the "description" field to the description itself.

5.7.2.4.2 Otherwise, the user must be asked for the description.

5.7.3 If the user must be asked for the input field to be changed, ask him for its name.

5.7.3.1 Print a question at the terminal.

"For which field do you want to change information?"

5.7.3.2 Wait for the user's response.

5.7.3.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter ? for an explanation,

Q to have the CHANGE command ignored,

COUNT to change a count or the units counted,

DEPTH to change depth information,

DESCRIPTION to change a specimen description,

GRID to change grid square information,

LOCATION to change general location information,

SITE to change site identification,

SPECIMEN to change specimen number,

UNIT to change excavation unit,

WEIGHT to change weight, or

YEAR to change year."

?, print an explanation at the terminal and repeat the question.

"You told SPECS to change specimen-catalog data that has already been entered. SPECS wants to know in which input field the information was entered."

SPECIMEN, GRID, UNIT, DEPTH, LOCATION, DESCRIPTION, COUNT, SITE, YEAR, or WEIGHT, set the "input field" value in the change-data record to the appropriate value from the "code" column of Table 16.

Q, print a message at the terminal and stop executing the CHANGE command.

"At your request, the CHANGE command is being ignored."

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your answer as the name of an input field."

5.7.3.4 If the field to be changed is unit, grid square, or depth, print a message at the terminal to warn the user that information stored in the "location" field is unaffected by the change.

In the message, field is replaced by the name of the input field.

"Field information that was included in "location" can be changed only by a CHANGE command for the "location" field."

5.7.4 If the user must be asked for the specimen number, ask him for it.

5.7.4.1 Print a prompt at the terminal.

"Please enter the specimen number:"

5.7.4.2 Wait for the user's response.

5.7.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CHANGE command ignored,

or

the specimen number, including accession number, of the specimen for which information is to be changed."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change specimen-catalog data that has already been entered. SPECS wants to know for which specimen the information was entered."

Q, print a message at the terminal and stop executing the CHANGE command.

"At your request, the CHANGE command is being ignored.

anything else, assume it is a specimen number.

5.7.4.4 Validate the assumed specimen number, using the algorithm in Appendix C. If validation fails, print a message at the terminal, then repeat the prompt for a specimen number.

"SPECS cannot interpret your answer as a specimen number."

5.7.4.5 If validation succeeded and the specimen number did not include an accession number, print a message at the terminal and repeat the prompt for a specimen number.

"The specimen number must include an accession number."

5.7.5 If the user must be asked for the description, ask him for it.

5.7.5.1 Print a prompt at the terminal.

"Please enter up to 1 line of description information:"

5.7.5.2 Wait for the user's response.

5.7.5.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter a part of the description of the specimen, in quotation marks,

NONE (without quotation marks) if you don't want to specify a description,

Q to have the CHANGE command ignored, or

? for an explanation."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change specimen-catalog data that has already been entered, and have attempted to give a part of the description of the specimens for which the information was entered. SPECS did not understand what you said about the description, and is asking you to repeat the information. Your answer will help SPECS recognize the specimen about which you want to change data: SPECS will change information only about specimens that have the correct specimen number and whose descriptions contain the value you provide."

Q, print a message at the terminal and stop executing the CHANGE command.

"At your request, the CHANGE command is being ignored."

a quoted character string, treat it as a description. That is, set the "description" field of the change-data record to the quoted string and the "transaction options" field to the number of characters in the string.

NONE, do nothing.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response as a description."

5.7.6 If the value of the "transaction options" field has not yet been set, set it to -1 to indicate the absence of instructions about multiple specimens with the same number.

5.7.7 Ask the user for the new value for the input field.

In the following messages to be printed at the terminal, field is replaced by "count," "depth," "description," "grid square," "location," "site," "specimen number," "unit," "weight," or "year."

5.7.7.1 Print a prompt at the terminal.

"Please enter new _field_ information:"

5.7.7.2 Wait for the user's response.

5.7.7.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CHANGE command ignored,

: to have this menu repeated,

NONE if there is no new value, or

a new value for the input field _field_."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change information from the _field_ input field. It wants to know with what new information you want to replace the old data from the field."

Q, print a message at the terminal and stop executing the CHANGE command.

"At your request, the CHANGE command is being ignored."

anything else (including NONE), assume it is the new value for the input field.

5.7.8 Print a session report entry summarizing the information in the command. (SRE 34)

5.7.9 Apply single-field validation for the new value, breaking it into fields that can be used in a specimen-data record.

5.7.9.1 If the value is for the "count" input field, use the algorithm in Appendix C.

5.7.9.1.1 Apply the algorithm.

5.7.9.1.2 If validation fails because the count is malformed, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 35)

"The new count you gave is not numeric, and SPECS cannot represent it. SPECS is ignoring your CHANGE command."

5.7.9.1.3 If the count is negative, print messages at the terminal and on the session report. (SRE 36)

"The count is negative, and is probably an error."

5.7.9.1.4 If the count was rounded to produce an integer, print messages at the terminal and on the session report. (SRE 37)

"The count was rounded to produce an integer, but is probably erroneous."

5.7.9.1.5 If the count was recoded as -32,768, print messages at the terminal and on the session report. (SRE 38)

"The count was too far negative to be represented, and has been changed to -32,768. This count is probably erroneous."

5.7.9.1.6 If the count was recoded as 32,767, print messages at the terminal and on the session report. (SRE 39)

"The count was too large to be represented, and has been changed to 32,767. This count is probably erroneous."

5.7.9.1.7 If the container count was too large, print messages at the terminal and on the session report. (SRE 40)

"The number of boxes, bags, vials, or jars is larger than the maximum number of containers allowed by the validation constants."

5.7.9.1.8 If the count type is "other," write a message at the terminal to suggest that the user may want to add information to the "description" field.

"SPECS cannot represent the kind of units you said were counted, and has encoded them as "other." If you wish, you may add more detailed information to the "description" field."

5.7.9.2 If the value is for the "depth" field, use the algorithm in Appendix C.

5.7.9.2.1 Apply the algorithm.

5.7.9.2.2 If the surface information cannot be represented, print messages at the terminal and on the session report. (SRE 41)

"SPECS cannot represent information about the depths of surface specimens. If you want

SPECS to retain more information than that the specimen was from the surface, add the extra information to the "location" input field."

5.7.9.2.3 If the level or zone number is malformed, print messages at the terminal and on the session report. (SRE 42)

"SPECS does not recognize the information you have given as a level or zone number, but will treat it as one anyway."

5.7.9.2.4 If the depth measurement was truncated, print messages at the terminal and on the session report. (SRE 43)

"The depth measurement you gave is too long to be represented, and is truncated to three characters. You may represent the depth more fully by adding information to the "location" input field."

5.7.9.2.5 If the level or zone number is too high, print messages at the terminal and on the session report. (SRE 44)

"The level or zone number is greater than the maximum level or zone number allowed by the validation constants."

5.7.9.2.6 If the depth type is "other," write a message at the terminal to tell the user that the depth information will be added to the "location" field.

"SPECS cannot represent the kind of depth information you have given, and has encoded it as "other." The depth information you gave will be added to the information from the "location" input field."

5.7.9.3 If the value is for the "description" field, use the algorithm in Appendix C.

5.7.9.3.1 Apply the algorithm.

5.7.9.3.2 If "disc" or "discs" was changed to "disk" or "disks", print messages at the terminal and on the session report. (SRE 45)

""disc" was changed to "disk" to standardize spelling."

5.7.9.3.3 If "C. S. P. P.", "CS PP", or "C.S.P.P." was changed to CSPP, print messages at the terminal and on the session report. (SRE 46)

In the messages to be printed at the terminal, <u>old spelling</u> is replaced by the spelling that was changed.

"<u>old spelling</u>" was changed to "CSPP" to standardize spelling.

5.7.9.3.4 If the description was truncated, print messages at the terminal and on the session report. (SRE 47)

"The description was truncated to 127 characters so that it could be represented."

5.7.9.3.5 Find out whether the new description information should replace, or be concatenated to, the old information.

5.7.9.3.5.1 Print a question at the terminal.

"Should the new value replace the old one or be added to it?"

5.7.9.3.5.2 Wait for the user's response.

5.7.9.3.5.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter REPLACE if the new value is to replace the old one,

ADD if the new value is to be added at the end of the old one,

Q to have the CHANGE command ignored, or

? for an explanation."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change a specimen's description, and have given SPECS the new description information. SPECS wants to know whether you want the old description of the specimen to be forgotten when it is replaced by the new description, or whether you want the old description to be changed by adding the new information to it."

REPLACE, remember that the new value replaces the old one.

ADD, remember that the new value is to be added to the old one.

Q, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 48)

"At your request, the CHANGE command is being ignored."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your response."

5.7.9.4 If the value is for the "grid square" input field, use the algorithm in Appendix C.

5.7.9.4.1 Apply the algorithm.

5.7.9.4.2 If the grid square or a coordinate is malformed, grid-square information may be erroneous or of an unexpected type. In this case, print messages at the terminal and on the session report. Ask the user to supply the information again, broken into its component parts.

5.7.9.4.2.1 Print messages at the terminal and on the session report to warn the user that the information could not be interpreted. (SRE 49)

"SPECS cannot interpret the grid square information. Please be sure that the information is correct, then repeat it as SPECS asks for it."

5.7.9.4.2.2 Find out what kind of information was given.

5.7.9.4.2.2.1 Print a question at the terminal.

"What kind of grid square information did you give?"

5.7.9.4.2.2.2 Wait for the user's response.

5.7.9.4.2.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter GRID for a grid square,

POINT for a point provenience,

AREA for an area denoted by a range of coordinates, NONE if you did not want to give grid square information,

OTHER if the information does not represent a grid square, point provenience, or range of coordinates,

Q to have the change command ignored, or

? for an explanation."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change information about a grid square. SPECS cannot interpret the new information you gave it, and is asking what kind of location the information described."

Q, print messages at the terminal and on the session report, and stop executing the CHANGE command.

(SRE 50)

"At your request, the CHANGE command is being ignored."

GRID, remember that the grid-square type is G.

POINT, remember that the grid-square type is P.

AREA, remember that the grid-square type is A.

NONE, remember that the grid square type is N.

OTHER, remember that the grid-square type is O.

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your response."

5.7.9.4.2.3 If the new grid square type is not N, ask the user for the N-S coordinate of the point that best represents the grid square information.

5.7.9.4.2.3.1 Print a prompt at the terminal.

"Please enter the N - S coordinate of the point that best represents the grid square information:"

5.7.9.4.2.3.2 Wait for the user's response.

5.7.9.4.2.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have your answer to the question about the grid square type ignored, or

a single integer representing the N-S coordinate of the point."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change information about a grid square. SPECS cannot interpret what you told it, and is asking you to repeat the information. It now wants the N-S coordinate (the number before the "L" or "R") of the single point on the site that best represents the grid square information you wanted to give."

Q, print a message at the terminal and repeat the question about the grid square type by restarting step 5.7.9.4.2.2.

"At your request, SPECS is ignoring your answer to the question about grid square type."

an integer, assume it is the N-S coordinate.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response."

5.7.9.4.2.4 If the new grid-square type is not N, ask the user for the "L" or "R" of the point that best represents the grid square.

5.7.9.4.2.4.1 Print a prompt at the terminal.

"Please enter the L or R of the point that best represents the grid square information:"

5.7.9.4.2.4.2 Wait for the user's response.

5.7.9.4.2.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter L if the point is west of the site datum,

R if the site is east of the datum,

? for an explanation, or

Q to have your answer to the question about grid square type ignored."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change information about a grid square. SPECS cannot interpret what you told it, and is asking you to repeat the information. It now wants the "L" or "R" from the coordinates of the point that best represents the grid square information."

L, remember that the point is in the left (west) half of the site.

R, remember that the point is in the right (east) half of the site.

Q, print a message at the terminal and repeat the question about the grid square type by restarting step 5.7.9.4.2.2.

"At your request, SPECS is ignoring your answer to the question about grid square type."

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response."

5.7.9.4.2.5 If the new grid square type is not N, ask the user for the E-W coordinate of the point that best represents the grid square information.

5.7.9.4.2.5.1 Print a prompt at the terminal.

"Please enter the E - W coordinate of the point that best represents the grid-square information:"

5.7.9.4.2.5.2 Wait for the user's response.

5.7.9.4.2.5.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have your answer to the question about grid square type ignored, or

a single integer representing the E-W coordinate of the point."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change information about a grid square. SPECS cannot interpret what you told it, and is asking you to repeat the information. It now wants the E - W coordinate (the number after the "L" or "R") of the single point on the site that best represents the grid square information you wanted to give."

Q, print a message at the terminal and repeat the question about the kind of grid square information by restarting step 5.7.9.4.2.2.

"At your request, SPECS is ignoring your answer to the question about grid square type."

an integer, assume it is the E-W coordinate.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response."

5.7.9.4.2.6 Apply the validation algorithm in Appendix C, recognizing that the grid square information is already divided into N-S and E-W coordinates and L or R half, and that the grid square type is known.

5.7.9.4.3 If the user was not asked to repeat the grid square information, he does not know what kind of information SPECS thinks it was given. Print a message at the terminal to tell him.

In this message, <u>type</u> is to be replaced by "area", "grid square", or "point provenience," as appropriate for the grid square type.

"The grid square information describes a <u>type</u>."

5.7.9.4.4 Write a session report entry about the kind of grid square information.

5.7.9.4.4.1 If the grid square type is N, write the entry. (SRE 51)

5.7.9.4.4.2 If the grid square type is A, write the entry. (SRE 52)

5.7.9.4.4.3 If the grid square type is O, write the entry. (SRE 53)

5.7.9.4.4.4 If the grid square type is G, write the entry. (SRE 54)

5.7.9.4.4.5 If the grid square type is P, write the entry. (SRE 55)

5.7.9.4.5 If validation for the current grid square information showed an that E - W coordinate is negative, write messages at the terminal and on the session report. (SRE 56)

"The east-west coordinate is negative, and is probably an error."

5.7.9.4.6 If a real-number E - W coordinate was converted to an integer, print messages at the terminal and on the session report. (SRE 57)

"The east-west coordinate was converted to an integer so that it could be represented."

5.7.9.4.7 If a real-number N - S coordinate was converted to an integer, print messages at the terminal and on the session report.

(SRE 58)

"The north-south coordinate was converted to an integer so that it could be represented."

5.7.9.4.8 If information about an area was converted to integers, print messages at the terminal and on the session report. (SRE 59)

"The area whose coordinates you gave is represented by the point at its southeast corner."

5.7.9.4.9 If an E - W coordinate for the left half of the site is not within the expected range, print messages at the terminal and on the session report. (SRE 60)

"The east-west coordinate is outside the range set by the validation constants for the left half of the site."

5.7.9.4.10 If an E - W coordinate for the right half of the site is not within the expected range, print messages at the terminal and on the session report. (SRE 61)

"The east-west coordinate is outside the range set by the validation constants for the right half of the site."

5.7.9.4.11 If the N - S coordinate is not within the expected range, print messages at the terminal and on the session report. (SRE 62)

"The north-south coordinate is outside the range set by the validation constants."

5.7.9.4.12 If the grid-square type is O, A, or P, the north-south and east-west coordinates do not represent all of the available grid-square information. Print a message at the terminal to tell the user that the full information will be added to the "location" field.

"The information you originally provided for grid square will be added to the information from the "location" input field. To change it, use a CHANGE command for the "location" field."

5.7.9.5 If the value is for the "location" field, use the algorithm in Appendix C.

5.7.9.5.1 Apply the algorithm.

5.7.9.5.2 If the information was truncated, print messages at the terminal and on the session report. (SRE 63)

"The location information was truncated to 127 characters so that it could be represented."

5.7.9.5.3 Find out whether the new location information should replace, or be concatenated to, the old information.

5.7.9.5.3.1 Print a question at the terminal.

"Should the new value replace the old one or be added to it?"

5.7.9.5.3.2 Wait for the user's response.

5.7.9.5.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter REPLACE if the new value is to replace the old one,

ADD if the new value is to be added at the end of the old one,

Q to have the CHANGE command ignored, or

? for an explanation."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to change location information about a specimen, and have given SPECS the new information. SPECS wants to know whether you want the old location information for the specimen to be forgotten when it is replaced by the new information, or whether you want the old location to be changed by adding the new information to it."

REPLACE, remember that the new value replaces the old one.

ADD, remember that the new value is to be added to the old one.

Q, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 48)

"At your request, the CHANGE command is being ignored."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your response."

5.7.9.6 If the value is for the "site" field, use the validation algorithm in Appendix C.

5.7.9.6.1 Apply the algorithm.

5.7.9.6.2 If validation fails because the site designation was missing, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 64)

"SPECS does not allow specimens to be represented without knowing their sites. Your request to remove the site information for the specimen is being ignored."

5.7.9.6.3 If validation fails because the site designation is too long, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 65)

"SPECS does not allow site designations with more than nine characters. Your request to change to a longer designation is being ignored."

5.7.9.7 If the value is for the "specimen number" input field, use the validation algorithm in Appendix C.

5.7.9.7.1 Apply the algorithm.

5.7.9.7.2 If validation fails, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 66)

"SPECS cannot interpret the new value as a specimen number, and will not execute the CHANGE command."

5.7.9.7.3 If the specimen number does not include an accession number, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 67)

"SPECS does not know the accession number for the new specimen number, and cannot execute the CHANGE command."

5.7.9.7.4 If a specimen type of "Bu" or "hb" was changed to "h", print messages at the terminal and on the session report. (SRE 68)

""Bu" and "hb" are changed to "h"."

5.7.9.7.5 If a specimen type of "eb" was changed to "e", print messages at the terminal and on the session report. (SRE 69)

""eb" is changed to "e"."

5.7.9.8 If the value is for the "unit" input field, use the algorithm in Appendix C.

5.7.9.8.1 Apply the algorithm.

5.7.9.8.2 If validation fails because a unit number is malformed, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 70)

In the message to be printed at the terminal, <u>type</u> is replaced by "feature", "burial", "post hole", "test pit", or "structure."

"SPECS cannot encode a <u>type</u> number that it is neither an integer nor an integer followed by a letter, and is ignoring your CHANGE command."

5.7.9.8.3 If the unit number included a letter, the unit information will be stored in two places: the numeric part of the designation will be stored in the "unit" field, and the abbreviated unit information produced during validation will be added to the "location" field. Print messages at the terminal and on the session report to tell the user about these places. (SRE 71)

In the message to be printed at the terminal, <u>number</u> represents the numeric part of the unit number, and <u>short</u> represents the abbreviated unit information.

"The unit identification will be represented both as <u>number</u> in the "unit" field of the specimen-data record and as <u>short</u> in the "location" field. The information in the "location" field may be altered by a CHANGE command for the "location" field."

5.7.9.8.4 If the unit type is "other", print a message at the terminal to tell the user that the unit information will be added to the "location" field.

"SPECS could not recognize the unit type, and has encoded it as "other." The information you provided will be added to the "location" field, and may be altered by a CHANGE command for the "location" field.

5.7.9.9 If the value is for the "weight" input field, use the validation algorithm in Appendix C.

5.7.9.9.1 Apply the algorithm.

5.7.9.9.2 If validation fails because the weight measurement is malformed, the information cannot be represented as a weight. Print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 72)

"SPECS cannot recognize the information you gave as a weight, and is ignoring your requested change. If you want SPECS to record this weight information, you may add it to the "description" input field."

5.7.9.9.3 If validation fails because the weight is not in ounces, the information should not be accepted as a weight. Print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 72)

"SPECS does not recognize the weight as in ounces, and is ignoring the CHANGE command. Please convert the weight to ounces and repeat the command."

5.7.9.9.4 If the weight was rounded, print messages at the terminal and on the session report. (SRE 73)

"The weight you gave was rounded to the nearest 0.1 ounce."

5.7.9.9.5 If the weight was changed to "> 3276.7", print messages at the terminal and on the session report. (SRE 74)

"The weight you gave was too large to represent, and is being recorded as greater than 3276.7 oz. (almost 205 lbs.). This weight is probably erroneous."

5.7.9.10 If the value is for the "year" input field, use the validation algorithm in Appendix C.

5.7.9.10.1 Apply the algorithm.

5.7.9.10.2 If the year was missing, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 75)

"SPECS does not allow specimens to be represented without year information. Your attempt to remove the year information for the specimen is being ignored."

5.7.9.10.3 If the year was incorrectly formatted, print messages at the terminal and on the session report, and stop executing the CHANGE command. (SRE 76)

"SPECS does not recognize the information you gave it as a year, and is ignoring your CHANGE command."

5.7.9.10.4 If the year is too early, print messages at the terminal and on the session report. (SRE 77)

"The year you have given is too early to be correct for any of the current RLA collections."

5.7.9.10.5 If the year is too late, print messages at the terminal and on the session report. (SRE 78)

"The year you have given has not yet arrived."

5.7.10 Use the original values and the values derived from validation to set the "input field," "field count," "field designation," "field size," "field format," and "new value" fields in the change-data record to be written for this CHANGE command.

5.7.10.1 If the input field was "count", set the change-data fields that affect the "count type" and "count" fields of the specimen-data record.

5.7.10.1.1 Set "input field" in the change-data record to 1, and set "field count" to 2.

5.7.10.1.2 If the type code is not N, set the change-data fields to reset the "count type" and "count" fields of the specimen-data record.

5.7.10.1.2.1 Create a new-field description to change "count type".

5.7.10.1.2.1.1 Set "field designation" to 9.

5.7.10.1.2.1.2 Set "field size" to 1.

5.7.10.1.2.1.3 Set "field format" to C.

5.7.10.1.2.1.4 Set "new value" to the type code from validation.

5.7.10.1.2.2 Create a new-field description to change "count".

5.7.10.1.2.2.1 Set "field designation" to 16.

5.7.10.1.2.2.2 Set "field size" to 2.

5.7.10.1.2.2.3 Set "field format" to N.

5.7.10.1.2.2.4 Set "new value" to the count determined by validation.

5.7.10.1.3 If the type code is N, set change-data fields to reset the "count type" field and eliminate the "count" field of the specimen-data record.

5.7.10.1.3.1 Create a new-field description to change "count type".

5.7.10.1.3.1.1 Set "field designation" to 9.

5.7.10.1.3.1.2 Set "field size" to 1.

5.7.10.1.3.1.3 Set "field format" to C.

5.7.10.1.3.1.4 Set "new value" to N.

5.7.10.1.3.2 Create a new-field description to eliminate "count".

5.7.10.1.3.2.1 Set "field designation" to 16.

5.7.10.1.3.2.2 Set "field size" to 0.

5.7.10.1.3.2.3 Set "field format" to N.

5.7.10.2 If the input field was "depth", set the change-data fields that affect the "depth type", "depth", and perhaps "location" fields of the specimen-data record. (If the "location" field is changed, the batch update will automatically reset the "location length.")

5.7.10.2.1 Set "input field" in the change-data record to 2.

5.7.10.2.2 If the depth type is N or S, set the change-data fields to reset the "depth type" field and eliminate the "depth" field of the specimen-data record.

  5.7.10.2.2.1 Set "field count" in the change-data record to 2.

  5.7.10.2.2.2 Create a new-field description to reset "depth type."

    5.7.10.2.2.2.1 Set "field designation" to 8.

    5.7.10.2.2.2.2 Set "field size" to 1.

    5.7.10.2.2.2.3 Set "field format" to C.

    5.7.10.2.2.2.4 Set "new value" to the depth type from validation (N or S).

  5.7.10.2.2.3 Create a new-field description to eliminate "depth."

5.7.10.2.2.3.1 Set "field designation" to 15.

5.7.10.2.2.3.2 Set "field size" to 0.

5.7.10.2.2.3.3 Set "field format" to C.

5.7.10.2.3 If the depth type from validation is 0, set the change-data fields to reset the "depth type" field, eliminate the "depth" field, and add to the "location" field of the specimen-data record.

5.7.10.2.3.1 Set "field count" in the change-data record to 3.

5.7.10.2.3.2 Create a new-field description to reset "depth type."

5.7.10.2.3.2.1 Set "field designation" to 8.

5.7.10.2.3.2.2 Set "field size" to 1.

5.7.10.2.3.2.3 Set "field format" to C.

5.7.10.2.3.2.4 Set "new value" to 0.

5.7.10.2.3.3 Create a new-field description to eliminate "depth."

5.7.10.2.3.3.1 Set "field designation" to 15.

5.7.10.2.3.3.2 Set "field size" to 0.

5.7.10.2.3.3.3 Set "field format" to C.

5.7.10.2.3.4 Create a new-field description to add to "location."

5.7.10.2.3.4.1 Set "field designation" to 20.

5.7.10.2.3.4.2 Set "field size" to the length of the original depth information given by the user.

5.7.10.2.3.4.3 Set "field format" to C.

5.7.10.2.3.4.4 Set "new value" to the original depth information given by the user.

5.7.10.2.4 If the depth type from validation is L or Z, set the change-data fields to reset the "depth type" and "depth" fields of the specimen-data record.

5.7.10.2.4.1 Set "field count" in the change-data record to 2.

5.7.10.2.4.2 Create a new-field description to reset "depth type."

5.7.10.2.4.2.1 Set "field designation" to 8.

5.7.10.2.4.2.2 Set "field size" to 1.

5.7.10.2.4.2.3 Set "field format" to C.

5.7.10.2.4.2.4 Set "new value" to the depth type from validation (L or Z).

5.7.10.2.4.3 Create a new-field description to reset "depth."

5.7.10.2.4.3.1 Set "field designation" to 15.

5.7.10.2.4.3.2 Set "field size" to 3.

5.7.10.2.4.3.3 Set "field format" to C.

5.7.10.2.4.3.4 Set "new value" to the depth information from validation.

5.7.10.3 If the input field was "description", set the change-data fields that affect the "description" field of the specimen-data record. (The "description length" field will be automatically reset by the batch update.)

5.7.10.3.1 Set "input field" in the change-data record to 3.

5.7.10.3.2 Set "field count" in the change-data record to 1.

5.7.10.3.3 If the new description information is to replace the old, create a new-field description.

5.7.10.3.3.1 Set "field designation" to 19.

5.7.10.3.3.2 Set "field size" to the length of the new description information,

5.7.10.3.3.3 Set "field format" to C.

5.7.10.3.3.4 Set "new value" to the new description information.

5.7.10.3.4 If the new description information is to be added to the old, create a new-field description.

5.7.10.3.4.1 Set "field designation" to 21.

5.7.10.3.4.2 Set "field size" to the length of the new description information.

5.7.10.3.4.3 Set "field format" to C.

5.7.10.3.4.4 Set "new value" to the new description information.

5.7.10.4 If the input field was "grid square", set the change-data fields that affect the "grid-square type," "N-S coord," "L-R half," and "E-W coord," and perhaps "location," fields of the specimen-data record. (If the "location" field is changed, the batch update will automatically reset the "location length.")

5.7.10.4.1 Set "input field" in the change-data record to 4.

5.7.10.4.2 If the grid-square type is N, set the change-data fields to reset the "grid-square type" field and eliminate the "E-W coord," "L-R half," and "N-S coord" fields of the specimen-data record.

5.7.10.4.2.1 Set the "field count" of the change-data record to 4.

5.7.10.4.2.2 Create a new-field description to reset "grid-square type."

5.7.10.4.2.2.1 Set "field designation" to 7.

5.7.10.4.2.2.2 Set "field size" to 1.

5.7.10.4.2.2.3 Set "field format" to C.

5.7.10.4.2.2.4 Set "new value" to N.

5.7.10.4.2.3 Create a new-field description to eliminate "N-S coord."

5.7.10.4.2.3.1 Set "field designation" to 12.

5.7.10.4.2.3.2 Set "field size" to 0.

5.7.10.4.2.3.3 Set "field format" to N.

5.7.10.4.2.4 Create a new-field description to eliminate "L-R half."

5.7.10.4.2.4.1 Set "field designation" to 13.

5.7.10.4.2.4.2 Set "field size" to 0.

5.7.10.4.2.4.3 Set "field format" to C.

5.7.10.4.2.5 Create a new-field description to eliminate "E-W coord."

5.7.10.4.2.5.1 Set "field designation" to 14.

5.7.10.4.2.5.2 Set "field size" to 0.

5.7.10.4.2.5.3 Set "field format" to N.

5.7.10.4.3 If the grid-square type is G, set the change-data fields to reset the "grid-square type," "N-S coord," "L-R half," and "E-W coord" fields of the specimen-data record.

  5.7.10.4.3.1 Set the "field count" of the change-data record to 4.

  5.7.10.4.3.2 Create a new-field description to reset "grid-square type."

    5.7.10.4.3.2.1 Set "field designation" to 7.

    5.7.10.4.3.2.2 Set "field size" to 1.

    5.7.10.4.3.2.3 Set "field format" to C.

    5.7.10.4.3.2.4 Set "new value" to G.

  5.7.10.4.3.3 Create a new-field description to reset "N-S coord."

    5.7.10.4.3.3.1 Set "field designation" to 12.

    5.7.10.4.3.3.2 Set "field size" to 2.

    5.7.10.4.3.3.3 Set "field format" to N.

5.7.10.4.3.3.4 Set "new value" to the north-south coordinate.

5.7.10.4.3.4 Create a new-field description to reset "L-R half."

5.7.10.4.3.4.1 Set "field designation" to 13.

5.7.10.4.3.4.2 Set "field size" to 1.

5.7.10.4.3.4.3 Set "field format" to C.

5.7.10.4.3.4.4 Set "new value" to the "L" or "R" indicating the half-site.

5.7.10.4.3.5 Create a new-field description to reset "E-W coord."

5.7.10.4.3.5.1 Set "field designation" to 14.

5.7.10.4.3.5.2 Set "field size" to 2.

5.7.10.4.3.5.3 Set "field format" to N.

5.7.10.4.3.5.4 Set "new value" to the east-west coordinate.

5.7.10.4.4 If the grid-square type is P, A, or O, set the change-data fields to reset the "grid-square type," "N-S coord," "L-R half," and "E-W coord" fields and add to the "location" field of the specimen-data record.

5.7.10.4.4.1 Set the "field count" of the change-data record to 5.

5.7.10.4.4.2 Create a new-field description to reset "grid-square type."

5.7.10.4.4.2.1 Set "field designation" to 7.

5.7.10.4.4.2.2 Set "field size" to 1.

5.7.10.4.4.2.3 Set "field format" to C.

5.7.10.4.4.2.4 Set "new value" to the grid-square type (P, A, or O).

5.7.10.4.4.3 Create a new-field description to reset "N-S coord."

5.7.10.4.4.3.1 Set "field designation" to 12.

5.7.10.4.4.3.2 Set "field size" to 2.

5.7.10.4.4.3.3 Set "field format" to N.

5.7.10.4.4.3.4 Set "new value" to the north-south coordinate.

5.7.10.4.4.4 Create a new-field description to reset "L-R half."

5.7.10.4.4.4.1 Set "field designation" to 13.

5.7.10.4.4.4.2 Set "field size" to 1.

5.7.10.4.4.4.3 Set "field format" to C.

5.7.10.4.4.4.4 Set "new value" to the "L" or "R" indicating the half-site.

5.7.10.4.4.5 Create a new-field description to reset "E-W coord."

5.7.10.4.4.5.1 Set "field designation" to 14.

5.7.10.4.4.5.2 Set "field size" to 2.

5.7.10.4.4.5.3 Set "field format" to N.

5.7.10.4.4.5.4 Set "new value" to the east-west coordinate

5.7.10.4.4.6 Create a new-field description to add to "location."

5.7.10.4.4.6.1 Set "field designation" to 20.

5.7.10.4.4.6.2 Set "field size" to the length of the new location information.

5.7.10.4.4.6.3 Set "field format" to C.

5.7.10.4.4.6.4 Set "new value" to the new location information.

5.7.10.5 If the input field was "location", set the change-data fields that affect the "location" field of the specimen-data record. (The "location length" field will be automatically reset by the batch update).

5.7.10.5.1 Set "input field" in the change-data record to 5.

5.7.10.5.2 Set "field count" in the change-data record to 1.

5.7.10.5.3 If the new information is to replace the old, create a new-field description to make the change.

5.7.10.5.3.1 Set "field designation" to 18.

5.7.10.5.3.2 Set "field size" to the length of the new information.

5.7.10.5.3.3 Set "field format" to C.

5.7.10.5.3.4 Set "new value" to the new location information.

5.7.10.5.4 If the new information is to be added to the old, create a new-field description to make the change.

5.7.10.5.4.1 Set "field designation" to 20.

5.7.10.5.4.2 Set "field size" to the length of the new information.

5.7.10.5.4.3 Set "field format" to C.

5.7.10.5.4.4 Set "new value" to the new location information.

5.7.10.6 If the input field was "site", set the change-data fields that affect the "site" field in the specimen-data record.

5.7.10.6.1 Set "input field" in the change-data record to 6.

5.7.10.6.2 Set "field count" in the change-data record to 1.

5.7.10.6.3 Create a new-field description to change "site".

5.7.10.6.3.1 Set "field designation" to 4.

5.7.10.6.3.2 Set "field size" to 9.

5.7.10.6.3.3 Set "field format" to C.

5.7.10.6.3.4 Set "new value" to the new site designation.

5.7.10.7 If the input field was "specimen number", set the change-data fields that affect the "accession number", "sequence and continuation number", and "specimen type" fields of the specimen-data record.

5.7.10.7.1 Set "input field" in the change-data record to 7.

5.7.10.7.2 Set "field count" in the change-data record to 3.

5.7.10.7.3 Create a new-field description to change "accession number".

5.7.10.7.3.1 Set "field designation" to 1.

5.7.10.7.3.2 Set "field size" to 2.

5.7.10.7.3.3 Set "field format" to N.

5.7.10.7.3.4 Set "new value" to the accession number from validation.

5.7.10.7.4 Create a new-field description to change "sequence and continuation number".

5.7.10.7.4.1 Set "field designation" to 2.

5.7.10.7.4.2 Set "field size" to 3.

5.7.10.7.4.3 Set "field format" to N.

5.7.10.7.4.4 Set "new value" to the sequence and continuation number from validation.

5.7.10.7.5 Create a new-field description to change "specimen type".

5.7.10.7.5.1 Set "field designation" to 3.

5.7.10.7.5.2 Set "field size" to 1.

5.7.10.7.5.3 Set "field format" to C.

5.7.10.7.5.4 Set "new value" to the specimen type code from validation.

5.7.10.8 If the input field was "unit", set the change-data fields that affect the "unit type," "unit", and perhaps "location" fields of the specimen-data record. (If the "location" field is changed, the batch update will automatically reset the "location length.")

5.7.10.8.1 Set "input field" in the change-data record to 8.

5.7.10.8.2 If the unit type from validation is N, set the change-data fields to reset the "unit type" field and eliminate the "unit" field of the specimen-data record.

  5.7.10.8.2.1 Set "field count" in the change-data record to 2.

  5.7.10.8.2.2 Create a new-field description to change "unit type".

    5.7.10.8.2.2.1 Set "field designation" to 6.

    5.7.10.8.2.2.2 Set "field size" to 1.

    5.7.10.8.2.2.3 Set "field format" to C.

    5.7.10.8.2.2.4 Set "new value" to N.

  5.7.10.8.2.3 Create a new-field description to eliminate "unit".

    5.7.10.8.2.3.1 Set "field designation" to 11.

    5.7.10.8.2.3.2 Set "field size" to 0.

    5.7.10.8.2.3.3 Set "field format" to N.

5.7.10.8.3 If the unit type from validation is 0, set the change-data fields to reset the "unit type", eliminate the "unit", and add to the "location" fields of the specimen-data record.

5.7.10.8.3.1 Set "field count" in the change-data record to 3.

5.7.10.8.3.2 Create a new-field description to reset "unit type".

5.7.10.8.3.2.1 Set "field designation" to 6.

5.7.10.8.3.2.2 Set "field size" to 1.

5.7.10.8.3.2.3 Set "field format" to C.

5.7.10.8.3.2.4 Set "new value" to 0.

5.7.10.8.3.3 Create a new-field description to eliminate "unit".

5.7.10.8.3.3.1 Set "field designation" to 11.

5.7.10.8.3.3.2 Set "field size" to 0.

5.7.10.8.3.3.3 Set "field format" to N.

5.7.10.8.3.4 Create a new-field description to add to "location".

5.7.10.8.3.4.1 Set "field designation" to 20.

5.7.10.8.3.4.2 Set "field size" to the length of the unit information given by the user.

5.7.10.8.3.4.3 Set "field format" to C.

5.7.10.8.3.4.4 Set "new value" to the unit information given by the user.

5.7.10.8.4 If the unit type from validation is F, B, P, T, or S and the unit description is entirely numeric, set the change-data fields to reset the "unit type" and "unit" fields of the specimen-data record.

5.7.10.8.4.1 Set "field count" in the change-data record to 2.

5.7.10.8.4.2 Create a new-field description to reset "unit type".

5.7.10.8.4.2.1 Set "field designation" to 6.

5.7.10.8.4.2.2 Set "field size" to 1.

5.7.10.8.4.2.3 Set "field format" to C.

5.7.10.8.4.2.4 Set "new value" to the unit type from validation (F, B, P, T, or S).

5.7.10.8.4.3 Create a new-field description to reset "unit".

5.7.10.8.4.3.1 Set "field designation" to 11.

5.7.10.8.4.3.2 Set "field size" to 2.

5.7.10.8.4.3.3 Set "field format" to N.

5.7.10.8.4.3.4 Set "new value" to the unit designation.

5.7.10.8.5 If the unit type from validation is F, B, P, T, or S and the unit designation includes a letter, set the change-data fields to reset the "unit type" and "unit" fields and add to the "location" field of the specimen-data record.

5.7.10.8.5.1 Set "field count" in the change-data record to 3.

5.7.10.8.5.2 Create a new-field description to reset "unit type".

5.7.10.8.5.2.1 Set "field designation" to 6.

5.7.10.8.5.2.2 Set "field size" to 1.

5.7.10.8.5.2.3 Set "field format" to C.

5.7.10.8.5.2.4 Set "new value" to the unit type from validation (F, B, P, T, or S).

5.7.10.8.5.3 Create a new-field description to reset "unit".

5.7.10.8.5.3.1 Set "field designation" to 11.

5.7.10.8.5.3.2 Set "field size" to 2.

5.7.10.8.5.3.3 Set "field format" to N.

5.7.10.8.5.3.4 Set "new value" to the numeric part of the unit designation.

5.7.10.8.5.4 Create a new-field description to add to "location".

5.7.10.8.5.4.1 Set "field designation" to 20.

5.7.10.8.5.4.2 Set "field size" to the length of the abbreviated full designation created during validation.

5.7.10.8.5.4.3 Set "field format" to C.

5.7.10.8.5.4.4 Set "new value" to the abbreviated unit information.

5.7.10.9 If the input field was "weight", set the change-data fields that affect the "weight type" and "weight" fields of the specimen-data record.

5.7.10.9.1 Set "input field" in the change-data record to 9.

5.7.10.9.2 Set "field count" in the change-data record to 2.

5.7.10.9.3 If the weight type from validation is N, set the change-data fields to reset the "weight type" field and eliminate the "weight" field of the specimen-data record.

5.7.10.9.3.1 Create a new-field description to reset "weight type".

5.7.10.9.3.1.1 Set "field designation" to 10.

5.7.10.9.3.1.2 Set "field size" to 1.

5.7.10.9.3.1.3 Set "field format" to C.

5.7.10.9.3.1.4 Set "new value" to N.

5.7.10.9.3.2 Create a new-field description to eliminate "weight".

5.7.10.9.3.2.1 Set "field designation" to 17.

5.7.10.9.3.2.2 Set "field size" to 0.

5.7.10.9.3.2.3 Set "field format" to N.

5.7.10.9.4 If the weight type from validation is anything other than N, set the change-data fields to reset the "weight type" and "weight" fields of the specimen-data record.

5.7.10.9.4.1 Create a new-field description to reset "weight type".

5.7.10.9.4.1.1 Set "field designation" to 10.

5.7.10.9.4.1.2 Set "field size" to 1.

5.7.10.9.4.1.3 Set "field format" to C.

5.7.10.9.4.1.4 Set "new value" to the weight type from validation (G, L, or E).

5.7.10.9.4.2 Create a new-field description to reset "weight".

5.7.10.9.4.2.1 Set "field designation" to 17.

5.7.10.9.4.2.2 Set "field size" to 2.

5.7.10.9.4.2.3 Set "field format" to N.

5.7.10.9.4.2.4 Set "new value" to the weight determined by validation.

5.7.10.10 If the input field was "year", set the change-data fields that affect the "year" field of the specimen-data record.

5.7.10.10.1 Set the "input field" of the change-data record to 10.

5.7.10.10.2 Set the "field count" of the change-data record to 1.

5.7.10.10.3 Create a new-field description to reset "year".

5.7.10.10.3.1 Set "field designation" to 5.

5.7.10.10.3.2 Set "field size" to 2.

5.7.10.10.3.3 Set "field format" to N.

5.7.10.10.3.4 Set "new value" to the year.

5.7.11 Write a change-data record on the transaction file, using these values:

Accession number, sequence and continuation number, and specimen type: values provided by the specimen number in the CHANGE command.

Transaction number: the next transaction number to be assigned.

Transaction type code: C.

Transaction options: the value provided during analysis of the CHANGE command.

Input field, field count: the values provided during processing of the new value.

Description: the value assigned during analysis of the CHANGE command. If no value was assigned, the field has none.

New-field descriptions: the values provided during processing of the new value.

5.7.12 Write session report entries describing the new record.

5.7.12.1 Write an introductory message. (SRE 79)

5.7.12.2 Describe the accession number, sequence and continuation number, specimen type, transaction number, transaction type code, input field, and field count. (SRE 80)

5.7.12.3 If the "transaction options" value is greater than 0, write a message about the description field. (SRE 81)

5.7.12.4 For each field in the new-field descriptions, write a session report entry.

5.7.12.4.1 If the field is "field designation," describe it. (SRE 82)

5.7.12.4.2 If the field is "field size," describe it. (SRE 83)

5.7.12.4.3 If the field is "field format," describe it. (SRE 84)

5.7.12.4.4 If the field is "new value," describe it. (SRE 85)

5.7.13 Increment the next transaction number to be assigned.

5.7.14 Write messages about the new change-data record at the terminal and on the session report.

In the messages to be written at the terminal, <u>input field</u> is replaced by "count", "depth", "description", "grid square", "location", "site", "specimen number", "unit", "weight", or "year". <u>New value</u> is replaced by the new value. <u>Spec#</u> is replaced by the number of the specimen to be changed. Trans# is replaced by the transaction number of the change-data record. Description is replaced by the description information given to recognize specimens to be changed, if any was given. [All] is omitted unless the CHANGE command included ALL. [That have description "description"] is omitted if no description information was given for recognizing specimens to be changed.

5.7.14.1 If something other than the specimen number was changed, tell the user about it.

5.7.14.1.1 Write an introductory message. (SRE 86)

"At the next batch update, information from the input field field will be changed for [all] specimens numbered spec# [that have description "description"]. (Transaction # trans#)"

5.7.14.1.2 Write a message at the terminal to tell the user about the new value.

5.7.14.1.2.1 If the field to be changed was "count", tell the user.

In this message, <u>count</u> is replaced by the count information given by the user, <u>type</u> is replaced by the count type from validation, and <u>number</u> is replaced by the count from validation.

"The new count information is <u>count</u>, which is encoded:

Count type: <u>count type</u>

Count: <u>number</u>"

5.7.14.1.2.2 If the field to be changed was "depth," tell the user.

In this message, <u>depth</u> is replaced by the depth information given by the user, <u>type</u> is replaced by the depth type from validation, and <u>depth ID</u> is replaced by the depth from validation.

5.7.14.1.2.2.1 Tell the user about the information that is not in the "location" field.

"The new depth information is <u>depth</u>, which is encoded:

Depth type: <u>type</u>

Depth: <u>depth ID</u>"

5.7.14.1.2.2.2 If the depth information was added to the "location" field, tell the user.

"<u>depth</u> was added to the "location" field."

5.7.14.1.2.3 If the field to be changed was "description," tell the user.

In this message, <u>new</u> <u>description</u> is replaced by the description information given by the user, and <u>length</u> is replaced by the description length from validation.

"The new description information is <u>new</u> <u>description</u>.
The description length is <u>length</u>."

5.7.14.1.2.4 If the field to be changed was "grid square," tell the user.

In this message, <u>grid</u> <u>square</u> is replaced by the grid square information given by the user, <u>type</u> is replaced by the grid square type from validation, <u>N-S</u> <u>coord</u> is replaced by the N-S coordinate from validation, <u>L-R</u> <u>half</u> is replaced by the L-

R half from validation, and <u>E-W</u> <u>coord</u> is replaced by the E-W coordinate from validation.

5.7.14.1.2.4.1 Tell the user about the information that is not in the "location" field.

"The new grid square information is <u>grid</u> <u>square</u>, which is encoded:

Grid square type: <u>type</u>

N - S coord: <u>N-S</u> <u>coord</u>

L - R half: <u>L-R</u> <u>half</u>

E - W coord: <u>E-W</u> <u>coord</u>"

5.7.14.1.2.4.2 If the grid square information was added to the "location" field, tell the user.

"<u>grid</u> <u>square</u> was added to the "location" field."

5.7.14.1.2.5 If the field to be changed was "location," tell the user.

In this message, <u>location</u> is replaced by the location information given by the user, and length is replaced by the location length from validation.

"The new location information is location. The location length is length."

5.7.14.1.2.6 If the field to be changed was "site," tell the user.

In this message, site is replaced by the site information given by the user.

"The new site information is site."

5.7.14.1.2.7 If the field to be changed was "unit," tell the user.

In this message, unit is replaced by the unit information given by the user, type is replaced by the unit type from validation, and unit ID is replaced by the unit from validation.

5.7.14.1.2.7.1 Tell the user about the information that is not in the "location" field.

"The new unit information is unit, which is encoded:

Unit type: type

Unit: unit ID"

5.7.14.1.2.2.2 If the unit information was added to the "location" field, tell the user.

"unit was added to the "location" field."

5.7.14.1.2.8 If the field to be changed was "weight", tell the user.

In this message, weight is replaced by the weight information given by the user, type is replaced by the weight type from validation, and ounces is replaced by the weight from validation.

"The new weight information is weight, which is encoded:

Weight type: weight type

Weight: ounces"

5.7.14.1.2.9 If the field to be changed was "year," tell the user.

In this message, <u>year</u> is replaced by the year information given by the user.

"The new year information is <u>year</u>."

5.7.14.2 If the specimen number was to be changed, tell the user about it.

5.7.14.2.1 Write introductory messages at the terminal and on the session report. (SRE 87)

"After TWO batch updates, the specimen number will be changed for [all] specimens currently numbered <u>spec#</u> [that have description "<u>description</u>".]
(Transaction # <u>trans#</u>)."

5.7.14.2.2 Tell the user about the new value.

In this message, <u>new spec#</u> is replaced by the specimen number given by the user, <u>acc#</u> is replaced by the accession number from validation, <u>type</u> is replaced by the specimen type from validation, and <u>seq#</u> is replaced by the sequence and continuation number from validation.

"The new specimen number is <u>new spec#</u>, which is encoded:

Accession number: <u>acc#</u>

Specimen type: <u>type</u>

Sequence and continuation number: <u>seq#</u>"

5.8 If the response to the request for a command was:

CHECK, write a validation-restoration record and repeat the request for a command.

5.8.1 If there is no more information in the command, the user must be asked for the transaction and specimen numbers of the transaction for which validation is to be restored.

5.8.2 If more information was given in the command, analyze it.

5.8.2.1 Assume the second word of the command (the one after CHECK) is a transaction number.

5.8.2.2 If the assumed transaction number is not an integer between 0 and the next transaction number to be assigned, the user must be asked for the transaction number.

5.8.2.3 If the assumed transaction number is the last word in the command, the user must be asked for the specimen number.

5.8.2.4 If there is more information in the command, analyze it to find the specimen number.

5.8.2.4.1 If the first word in the rest of the command is FOR, assume everything following it is a specimen number.

5.8.2.4.2 If the first word is not FOR, remember to ask the user for a specimen number.

5.8.2.4.3 Validate the assumed specimen number with the algorithm in Appendix C. If the validation fails, or if the specimen number has no accession number, the user must be asked for a specimen number.

5.8.3 If earlier processing shows it is needed, ask the user for the transaction number.

5.8.3.1 Print a prompt at the terminal.

"Please enter the transaction number:"

5.8.3.2 Wait for the user's response.

5.8.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CHECK command ignored,
or

the number of the transaction whose
validation should be restored."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to restore batch-update validation for a transaction in the current transaction file. SPECS wants to know the number of the transaction for which validation is to be restored."

Q, print a message at the terminal, and stop executing the CHECK command.

"At your request, the CHECK command is being ignored.

an integer betwen 0 and the next transaction number to be assigned, consider it the transaction number.

anything else, print a message at the terminal and repeat the prompt.

"SPECS does not recognize your response as a transaction number."

5.8.4 If earlier processing shows that it is needed, ask the user for the specimen number.

5.8.4.1 Print a prompt at the terminal.

"Please enter the specimen number:"

5.8.4.2 Wait for the user's response.

5.8.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CHECK command ignored, or

the specimen number, including accession number, of the specimen that will be affected by the transaction whose validation should be restored."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to restore batch-update validation for a transaction in the current transaction file. SPECS wants to know the number of the specimen that will be affected by the transaction whose validation is restored."

Q, print a message at the terminal, and stop executing the CHECK command.

"At your request, the CHECK command is being ignored."

anything else, assume it is a specimen number.

5.8.4.4 Validate the assumed specimen number, using the algorithm in Appendix C. If validation fails, print a message at the terminal, then repeat the request for a specimen number by restarting step 5.8.4.

"SPECS does not recognize your response as a specimen number."

5.8.4.5 If the validation did not fail and the specimen number did not include an accession number, print a message at the terminal, then repeat the request for a specimen number by restarting step 5.8.4.

"The specimen number must include an accession number."

5.8.5 Write a session report entry summarizing the information in the command. (SRE 88)

5.8.6 Write a validation-restoration record on the transaction file, using these values:

Accession number, sequence and continuation number, and specimen type: from validation of the specimen number.

Transaction number: the next transaction number to be assigned.

Transaction type code: R

Affected transaction number: the number of the transaction for which validation is to be restored.

5.8.7 Increment the next transaction number to be assigned.

5.8.8 Write messages about the validation-restoration record at the terminal and on the session report. (SRE 89)

In the message to be written at the terminal, restore# is replaced by the number of the transaction whose validation is to be restored, spec# is replaced by the specimen number, and trans# is replaced by the transaction number of the validation-restoration record.

"Validation for transaction # restore#, referring to specimen spec#, will be restored during the next batch update.
(Transaction # trans#)."

5.9 If the response to the request for a command was:

COMMENT, write a comment on the session report and repeat the request for a command.

5.9.1 If COMMENT is not the last word in the command, consider the rest of the command a comment.

5.9.2 Otherwise, ask the user for the comment.

5.9.2.1 Print a prompt at the terminal.

"Please enter the comment:"

5.9.2.2 Wait for the user's response.

5.9.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

: to have this menu repeated,

Q to request that your COMMENT command be ignored, or

any comment, up to 1 line long."

?, print an explanation at the terminal and repeat the prompt.

"You told SPECS to write a comment on the session report. SPECS wants to know what to write."

Q, print a message at the terminal and stop executing the COMMENT command.

"At your request, SPECS is ignoring the COMMENT command."

anything else, consider it the comment.

5.9.3 Write the comment on the session report, prefaced by "USER'S COMMENT:" (SRE 90)

5.10 If the response to the request for a command was:

CONFIRM, write a validation-override record and repeat the request for a command.

5.10.1 If there is no more information in the command, the user must be asked for the transaction and specimen numbers of the transaction for which validation is to be overridden.

5.10.2 If more information was given in the command, analyze it.

  5.10.2.1 Assume that the second word in the command (the one after CONFIRM) is a transaction number.

  5.10.2.2 If the assumed transaction number is not an integer between 0 and the next transaction number to be assigned, the user must be asked for the transaction number.

  5.10.2.3 If the assumed transaction number is the last word in the command, the user must be asked for the specimen number.

  5.10.2.4 If there is more information in the command, analyze it to find the specimen number.

    5.10.2.4.1 If the first word in the rest of the command is FOR, assume everything following it is a specimen number.

    5.10.2.4.2 If the first word is not FOR, the user must be asked for a specimen number.

5.10.2.4.3 Validate the assumed specimen number with the algorithm in Appendix C. If the validation fails, or if the specimen number has no accession number, the user must be asked for a specimen number.

5.10.3 If earlier processing shows it is needed, ask the user for the transaction number.

5.10.3.1 Print a prompt at the terminal.

"Please enter the transaction number:"

5.10.3.2 Wait for the user's response.

5.10.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CONFIRM command ignored, or

the transaction number of the transaction whose validation should be overridden."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to override batch-update validation for a transaction in the current transaction file. SPECS wants to know the number of the transaction whose validation is to be overridden."

Q, print a message at the terminal, and stop executing the CONFIRM command.

"At your request, the CONFIRM command is being ignored.

an integer betwen 0 and the next transaction number to be assigned, consider it the transaction number.

anything else, print a message at the terminal and repeat the prompt.

"SPECS does not recognize your response as a transaction number."

5.10.4 If earlier processing shows that it is needed, ask the user for the specimen number.

5.10.4.1 Print a prompt at the terminal.

"Please enter the specimen number:"

5.10.4.2 Wait for the user's response.

5.10.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the CONFIRM command ignored, or

the specimen number, including accession number, of the specimen that will be affected by the transaction for which validation should be overridden."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to override batch-update validation for a transaction in the current transaction file. SPECS wants to know the number of the specimen that will be affected by the transaction whose validation is overridden."

Q, print a message at the terminal, and stop executing the CONFIRM command.

"At your request, the CONFIRM command is being ignored."

anything else, assume it is a specimen number.

5.10.4.4 Validate the assumed specimen number, using the algorithm in Appendix C. If validation fails, print a message at the terminal and repeat the request for a specimen number by restarting step 5.10.4.

"SPECS does not recognize your response as a specimen number."

5.10.4.5 If the validation succeeded and the specimen number does not include an accession number, print a message at the terminal, then repeat the request for a specimen number by restarting step 5.10.4

"The specimen number must include an accession number."

5.10.5 Write a session report entry summarizing the information in the command. (SRE 91)

5.10.6 Write a validation-override record on the transaction file, using these values:

Accession number, sequence and continuation number, and specimen type: from validation of the specimen number.

Transaction number: the next transaction number to be assigned.

Transaction type code: 0

Affected transaction number: the number of the transaction for which validation is to be overridden.

5.10.7 Increment the next transaction number to be assigned.

5.10.8 Write messages about the validation-override record at the terminal and on the session report. (SRE 92)

In the message to be written at the terminal, <u>confirm#</u> is replaced by the number of the transaction whose validation is to be overridden, <u>spec#</u> is replaced by the specimen number, and <u>trans#</u> is replaced by the transaction number of the validation-override record.

"Validation for transaction # <u>confirm#</u>, referring to specimen <u>spec#</u>, will be overridden during the next batch update.
(Transaction # <u>trans#</u>)."

5.11 If the response to the request for a command was:

DELETE, write a delete-data record and repeat the request for a command.

5.11.1 If there is no more information in the command, the user must be asked for the specimen number and description of the record to be deleted.

5.11.2 If more information was given in the command, analyze it.

5.11.2.1 Consider the second word of the command (the one after DELETE).

5.11.2.2 If the second word is ALL, set the "transaction options" field of the delete-data record to 0. Get the next word of the command.

5.11.2.3 The current word is assumed to be a specimen number. Validate it with the algorithm in Appendix C.

5.11.2.4 If the validation failed, or the specimen number had no accession number, the user must be asked for the specimen number and description of the record to be deleted.

5.11.2.5 If the validation did not fail and the rest of the command is WITH followed by a character string enclosed in quotation marks, the command includes a description. In this case, set the "description" field of the delete-data record to the character string and the "transaction options" field to the length of the character string.

5.11.2.6 If the specimen-number validation did not fail and the specimen number was followed by at least one character that was not the WITH and character string, the description is garbled. In this case, the user must be asked for the description.

5.11.3 If earlier processing shows that the user must be asked for the specimen number, ask him for it.

5.11.3.1 Print a prompt at the terminal.

"Please enter the specimen number:"

5.11.3.2 Wait for the user's response.

5.11.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the DELETE command ignored,
or

the specimen number, including accession number, of the specimen to be deleted."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS to delete all specimen-catalog data about a specimen; it wants to know which one."

Q, print a message at the terminal and stop executing the DELETE command.

"At your request, SPECS is ignoring the DELETE command."

anything else, assume it is a specimen number.

5.11.3.4 Validate the assumed specimen number using the algorithm in Appendix C. If the validation fails, print a message at the terminal, then ask the user for another specimen number by restarting step 5.11.3.

"SPECS cannot interpret your answer as a specimen number."

5.11.3.5 If the validation succeeded but the specimen number did not include an accession number, print a message at the terminal, then repeat the request for a specimen number by restarting step 5.11.3.

"The specimen number must include an accession number."

5.11.4 If earlier processing shows that the user must be asked for the description, ask him for it.

5.11.4.1 Print a prompt at the terminal.

"Please enter up to 1 line of description:"

5.11.4.2 Wait for the user's response.

5.11.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter a part of the description of the
specimen, in quotation marks,

NONE (without quotation marks) if you
don't want to specify a
description,

Q to have the DELETE command ignored,
or

? for an explanation."

?, print an explanation at the terminal and repeat the prompt.

"You told SPECS to delete all specimen-catalog data about a specimen, and tried to tell SPECS a part of the specimen's description. SPECS did not understand what you said about the description, and is asking you to repeat the information. Your answer will help SPECS recognize the specimen for which you want to delete information: SPECS will delete information only about specimens that have the correct specimen number and whose descriptions contain the value you provide now."

Q, print a message at the terminal and stop executing the DELETE command.

"At your request, SPECS is ignoring the DELETE command."

a quoted character string, treat it as a description. That is, set the "description" field of the delete-data record to the quoted string and the "transaction options" field to the number of characters in the string.

NONE, do nothing.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response as a description."

5.11.5 If the "transaction options" field in the delete-data record has no value, no instructions were given for processing multiple records with the same specimen number during the batch update. In this case, set the "transaction options" field to -1 to signal the absence of other instructions.

5.11.6 Print an entry about the command on the session report. (SEE 93)

5.11.7 Write a delete-data record on the transaction file, using these values:

Accession number, sequence and continuation number, and specimen type: from validation of the specimen number.

Transaction number: the next transaction number to be assigned.

Transaction type code: D

Transaction options: the value set while processing the DELETE command.

Description: the value set while processing the DELETE command. If no value was set, the "description" field has no value.

5.11.8 Increment the next transaction number to be assigned.

5.11.9 Write messages about the new delete-data record at the terminal and on the session report. (SRE 94)

In the following message, [all] is omitted unless the DELETE command included ALL; <u>spec#</u> is replaced by the number of the specimen to be deleted; [that have description "<u>description</u>"] is omitted unless description information was given; _description_ is replaced by the description information; and _trans#_ is replaced by the transaction number of the delete-data record.

"At the next batch update, [all] specimens numbered _spec#_ [that have description "_description_"] will be deleted.
(Transaction _trans#_).

5.12 If the response to the request for a command was:

END, print messages at the terminal and on the session report, and stop interpreting commands (step 5). (SEE 95)

"SPECS has finished asking you for commands."

5.13 If the response to the request for a command was:

INPUT, accept information and write insert-data records for one or more new specimens.

5.13.1 Prepare for information about the first new specimen.

5.13.1.1 Remember that there is no "previous specimen."

5.13.1.2 Print input headers at the terminal. (Terminal Report 4)

5.13.2 Execute the INPUT loop until a Q command is received instead of information about a new specimen.

5.13.2.1 Print a prompt at the terminal.

"Please enter information about a new specimen:"

5.13.2.2 Wait for the user's response.

5.13.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

HEADER to have SPECS repeat the column headings,

Q if you do not want to give information about another specimen, or

information about a new specimen."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS that you want to give it information about new specimens. It wants you to give it the information from the specimen catalog entry for a single specimen."

HEADER, print the column headings at the terminal and repeat the prompt. (Terminal Report 4)

Q, print a message at the terminal and stop executing the INPUT loop (step 5.13.2).

"At your request, SPECS is finishing the INPUT command."

anything else, assume it is information about a new specimen.

5.13.2.4 Process the information about a single new specimen.

5.13.2.4.1 Get all of the information about the specimen. To do this, repeat steps 5.13.2.4.1.1 through 5.13.2.4.1.3 until the last line of information about the specimen is found.

5.13.2.4.1.1 Wait for the user to enter a line of input.

5.13.2.4.1.2 If the line has at least one field that is continued (that is, it has a

"+" followed by either two or more blanks or a carriage return), it is not the last line for the specimen.

5.13.2.4.1.3 Otherwise, the line is the last one for the specimen.

5.13.2.4.2 Divide the new information into values for the input fields.

5.13.2.4.2.1 Identify the input fields in the first line.

5.13.2.4.2.1.1 Break the line into "spec-line fields," each containing all of the information from a single input field. To do this, repeat steps 5.13.2.4.2.1.1.1 to 5.13.2.4.2.1.1.4 until the last spec-line field is found.

5.13.2.4.2.1.1.1 Ignore any blanks at the beginning of the part of the line in which the spec-line fields have not yet been found.

5.13.2.4.2.1.1.2 Consider all characters before the next carriage return or the next character preceded by two or more blanks to be a spec-line field.

5.13.2.4.2.1.1.3 Remove any blanks at the end of the spec-line field.

5.13.2.4.2.1.1.4 If the spec-line field ends with a carriage return, it is the last field in the line.

5.13.2.4.2.1.2 If there are 8 spec-line fields, consider them the values of the 8 input fields expected in the first line.

5.13.2.4.2.1.3 If there are not 8 input fields, ask the user which input field each spec-line field belongs to. Repeat this step once for each spec-line field, using them in left-to-right order of their positions in the input line.

In the messages printed at the terminal during this step, <u>field</u> is replaced by the spec-line field.

5.13.2.4.2.1.3.1 Print a question at the terminal.

"To which input field does <u>field</u> belong?"

5.13.2.4.2.1.3.2 Wait for the user's response.

5.13.2.4.2.1.3.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter Q to have all of the information for this specimen ignored,

NONE to have the information that was printed in the question ignored,

? for an explanation, or

SPECIMEN,

GRID,

UNIT,

DEPTH,

LOCATION,

COUNT,

WEIGHT, or

DESCRIPTION."

?, print an explanation at the terminal and repeat the question.

"SPECS thinks you are telling it about a new specimen. It does not know how the information you have given corresponds to the input fields it expects, and is asking your help."

SPECIMEN, GRID, UNIT, DEPTH, LOCATION, DESCRIPTION, COUNT, or WEIGHT, remember which input field the information belongs to.

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information you gave for this specimen."

NONE, print a message at the terminal and ignore the spec-line field.

"<u>Field</u>" will be ignored."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer as the name of an input field."

5.13.2.4.2.1.4 If the values for the input fields were given in an unexpected order, or if the spec-line fields attributed to an input field were not adjacent, the line is scrambled. Tell the user, then repeat the request for information about a specimen by restarting step 5.13.2. (The expected order for input fields is: specimen number, grid square, unit, depth, location, count, weight, description. Some of these fields may not yet have values, and others may have several adjacent spec-line fields attributed to them.)

"The information for this specimen appears scrambled. Please re-enter the input, being sure that the fields are in the correct order."

5.13.2.4.2.1.5 If more than one adjacent spec-line field was attributed to the same input field, the breaks between the fields are probably unintended. Concatenate the spec-line fields belonging to each input field in left-to-right order, separated by single blanks, to produce the current value of the input field.

5.13.2.4.2.1.6 If there are input fields for which there are no spec-line fields, ask the user for their values by performing steps 5.14.2.4.2.1.6.1 through 5.14.2.4.2.1.6.3 for each such field.

In the messages printed at the terminal during this step, field is replaced by the name of the input field ("specimen number," "grid," "unit," "depth," "location," "count," "weight," or "description.")

5.13.2.4.2.1.6.1 Print a prompt at the terminal.

"Please enter a value for field:"

5.13.2.4.2.1.6.2 Wait for the user's response.

5.13.2.4.2.1.6.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter : to repeat this menu,

? for an explanation,

Q to have the information from this specimen ignored,

NONE if the field has no value for this specimen, or

the value of the field."

?, print an explanation at the terminal and repeat the prompt.

"SPECS thinks you are telling it about a new specimen. It cannot find the information for field."

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information you gave it about this specimen."

anything else (including NONE), consider it the value for the input field.

5.13.2.4.2.2 Divide the information in all lines after the first among the input fields, combining it with the information from the first line. Do this by performing steps 5.13.2.4.2.2.1 through 5.13.2.4.2.2.5 for each of the input lines.

5.13.2.4.2.2.1 Count the input fields (excluding those for which SPECS prompted the user) for which the current line is expected to provide continuations. These input fields have values that end with a "+".

5.13.2.4.2.2.2 Use the logic of step 5.13.2.4.2.1.1 to break the input line into spec-line fields.

5.13.2.4.2.2.3 If the number of spec-line fields is the same as the number of input fields for which continuations are expected, proceed in left-to-right order combining the spec-line fields with the input fields that need continuations. To combine a spec-line field and an input field, delete the "+" at the end of the input field, add a blank to the input field if it then ends in a non-blank character, and concatenate the spec-line field to the end of the input field.

5.13.2.4.2.2.4 If the number of spec-line fields differs from the number of input fields needing continuations, ask the user which input field each spec-line field belongs to. Repeat this step for each spec-line field, using the fields in left-to-right order of their appearances on the input line.

In the messages printed at the terminal during this step, field is replaced by the spec-line field.

5.13.2.4.2.2.4.1 Print a question at the terminal.

"To which input field does "field" belong?"

5.13.2.4.2.2.4.2 Wait for the user's response.

5.13.2.4.2.2.4.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter Q to have all of the information for this specimen ignored,

NONE to have the information printed in the question ignored,

? for an explanation,

SPECIMEN,

GRID,

UNIT,

DEPTH,

LOCATION,

COUNT,

WEIGHT, or

DESCRIPTION."

?, print an explanation at the terminal and repeat the question.

"SPECS thinks you are telling it about a new specimen. It does not know how the information you have given corresponds to the input fields it expects, and is asking your help."

SPECIMEN, GRID, UNIT, DEPTH, LOCATION, DESCRIPTION, COUNT, or WEIGHT, remember which input field the information belongs to.

Q, print a message at the terminal, and stop processing information for this specimen.

"At your request, SPECS will ignore the information you gave for this specimen."

NONE, print a message at the terminal, and ignore the spec-line field.

"At your request, field will be ignored."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

5.13.2.4.2.2.4.4 Combine the spec-line field with the current value of the input field. To combine the fields, remove the "+" that may be at the end of the input field, add a blank to the input field if it ends in a non-blank character, and concatenate the spec-line field to the end of the input field.

5.13.2.4.2.2.5 If there are input fields for which continuations were expected but not found, ask the user for the rest of the values for the fields. Repeat this step for each input field for which a continuation is missing.

In the messages printed at the terminal during this step, _field_ is replaced by the name of an input field and _value_ is replaced by the current value of the field.

5.13.2.4.2.2.5.1 Delete the "+" at the end of the input field. If the field does not then end in a blank, concatenate a single blank to its end.

5.13.2.4.2.2.5.2 Print a question at the terminal.

""_Value_" has been found as the information for _field_. What is the rest of the information?"

5.13.2.4.2.2.5.3 Wait for the user's response.

5.13.2.4.2.2.5.4 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter NONE if the value is complete,

: to have this menu repeated,

Q to have information from this specimen ignored,

? for an explanation, or

the rest of the information for <u>field</u>."

?, print an explanation at the terminal and repeat the question.

"SPECS thinks you are telling it about a new specimen. You have indicated that information from the <u>field</u> field will be found on more than one line. SPECS does not think it has found all of the information you gave it, and is asking you to tell it what it is missing."

NONE, consider the input field complete.

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else, delete any leading blanks from the response, then concatenate the response to the end of the value for the input field.

5.13.2.4.3 Find a value for each input field whose value is a single ditto mark (").

5.13.2.4.3.1 If there is a "previous specimen," replace the ditto mark by the value of the same input field for the previous specimen. Print a message at the terminal and remember (but do not write) one for the session report. (SRE 96)

In the message at the terminal, <u>field</u> is replaced by the name of the input field,

<u>value</u> is replaced by the field's new value, and <u>old spec#</u> is the specimen number of the "previous specimen."

"The <u>field</u> information is "<u>value</u>." It was copied from specimen <u>old spec#</u>."

5.13.2.4.3.2 If there is no "previous specimen," ask the user for the value of the input field.

In this step, <u>field</u> is replaced by the name of the input field.

5.13.2.4.3.2.1 Print a prompt at the terminal.

"Please enter the information from the <u>field</u> field:"

5.13.2.4.3.2.2 Wait for the user's response.

5.13.2.4.3.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter NONE if there is no value,

Q to have information for this specimen ignored,

? for an explanation,

: to have this menu repeated, or

the information."

?, print an explanation at the terminal and repeat the prompt.

"SPECS thinks you are telling it about a new specimen. You have told it that the information about the <u>field</u> is the same as it was in the last specimen. SPECS does not remember which was the last specimen, and wants you to repeat the information."

Q, print a message at the terminal and stop processing information for the specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else (including NONE), treat it as the value for the input field.

5.13.2.4.4 Validate the specimen number.

5.13.2.4.4.1 Apply the algorithm in Appendix C.

5.13.2.4.4.2 If validation fails, print an error message at the terminal, then get a new specimen number and repeat the validation.

In the messages printed at the terminal during this step, <u>value</u> is replaced by the specimen number.

5.13.2.4.4.2.1 Write a message at the terminal to tell the user that validation failed.

"SPECS cannot recognize <u>value</u> as a specimen number."

5.13.2.4.4.2.2 Print a prompt at the terminal to ask for a new value.

"Please enter the specimen number:"

5.13.2.4.4.2.3 Wait for the user's response.

5.13.2.4.4.2.4 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to have information for
this specimen ignored,

? for an explanation, or

a specimen number."

?, print an explanation at the terminal and repeat the prompt.

"SPECS thinks you are telling it about a new specimen. It could not recognize the specimen number you gave, and wants you to give the correct specimen number."

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else, consider it a specimen number and restart step 5.13.2.4.4.

5.13.2.4.4.3 If a specimen type of "Bu" or "hb" was recoded, print a message at the terminal and remember (but do not write) one for the session report. (SRE 68)

""Bu" and "hb" are recorded as "h"."

5.13.2.4.4.4 If a specimen type of "eb" was recoded, print a message at the terminal and remember (but do not write) one for the session report. (SRE 69)

""eb" is recorded as "e"."

5.13.2.4.5 If the accession number was not given as part of the specimen number and there is a "previous specimen," assume that the current specimen's site, year, and accession number are the same as those of the previous specimen. Print a message at the terminal and remember (but do not write) one for the session report. (SRE 97)

In the message at the terminal, <u>site</u> is replaced by the site, <u>year</u> is replaced by the year, and <u>acc#</u> is replaced by the accession number.

"The site, year, and accession number are assumed to be <u>site</u>, <u>year</u>, and <u>acc#</u>, as they were for the last specimen."

5.13.2.4.6 If the accession number for the current specimen is not yet known, ask the user for it.

5.13.2.4.6.1 Print a prompt at the terminal.

"Please enter the accession number:"

5.13.2.4.6.2 Wait for the user's response.

5.13.2.4.6.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to have information for this specimen ignored,

? for an explanation, or

an accession number."

?, print an explanation at the terminal and repeat the prompt.

"SPECS thinks you are telling it about a new specimen. The specimen number you gave does not have an accession number. SPECS assumes that the accession number has not changed since the last specimen for which you gave an accession number, but cannot recall which specimen that was."

Q, print a message at the terminal, and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else, assume it is an accession number.

5.13.2.4.6.4 Use the algorithm in Appendix C to validate the assumed accession number. If validation fails, print a message at the terminal and get another accession number by restarting step 5.13.2.4.6.

"SPECS cannot interpret your response as an accession number."

5.13.2.4.7 If the accession number was given, either as part of the specimen number or in response to a prompt, ask the user for the site at which the specimen was collected.

5.13.2.4.7.1 Print a prompt at the terminal.

"Please enter the site:"

5.13.2.4.7.2 Wait for the user's response.

5.13.2.4.7.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to have information for this specimen ignored,

? for an explanation, or

the site at which the specimen was collected."

?, print an explanation at the terminal and repeat the prompt.

"SPECS thinks you are telling it about a new specimen, for which you gave an accession number. SPECS is asking for the site and year in which the specimen was collected. It will both record the site and year you give it and use them to check the accession number."

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else, assume it is the site.

5.13.2.4.7.4 Use the algorithm in Appendix C to validate the assumed site information. If the validation fails, print a message at the terminal and get another site by restarting step 5.13.2.4.7.

"SPECS cannot interpret your response as a site."

5.13.2.4.8 If the accession number was given, ask the user for the year during which the specimen was collected.

5.13.2.4.8.1 Print a prompt at the terminal.

"Please enter the year:"

5.13.2.4.8.2 Wait for the user's response.

5.13.2.4.8.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q to have information for this specimen ignored,

? for an explanation, or

the year during which the specimen was collected."

?, print an explanation at the terminal and repeat the prompt.

"SPECS thinks you are telling it about a new specimen, for which you gave an accession number. SPECS is asking for the site and year in which the specimen was collected. It will both record the site and year you give and use them to check the accession number."

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else, assume it is the year.

5.13.2.4.8.4 Use the algorithm in Appendix C to validate the assumed year information.

If the validation fails, print a message at the terminal and get another year by restarting step 5.13.2.4.8.

"SPECS cannot interpret your response as a year."

5.13.2.4.9 If there is no "previous specimen," or if the site, year and accession number of the current specimen are not the same as those of the previous specimen, perform site-yr-acc validation. If the validation fails, tell the user why.

In the messages printed at the terminal during this step, <u>other acc#</u> is replaced by a list of the accession numbers that the validation found were already in the site-yr-acc file for the site and year. <u>Acc#</u> is replaced by the current accession number, <u>site</u> is replaced by the site, and <u>year</u> is replaced by the year.

5.13.2.4.9.1 Apply the validation algorithm in Appendix C.

5.13.2.4.9.2 If validation fails because the site and year are not entered in the site-yr-acc file, print a message at the terminal.

"Accession number acc# is not recorded for collections made during year at site."

5.13.2.4.9.3 If validation fails because the accession number is inconsistent with previous entries in the site-yr-acc file, print a message at the terminal.

"Accession number(s) other acc# are recorded for collections made during year at site, but acc# is not."

5.13.2.4.10 If the site-yr-acc validation failed, ask the user whether the site, year, and accession number are correct. If they are, enter them in the site-yr-acc file, supplementing other entries for the same site and year. If the site, year, and accession number are not correct, ask the user for a new site, year, and accession number.

5.13.2.4.10.1 Print a question at the terminal. In the question, site, year and acc# are replaced by the site, year, and accession number.

"Are site, year and acc# the correct site, year, and accession number for this specimen?"

5.13.2.4.10.2 Wait for the user's response.

5.13.2.4.10.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter YES if the site, year, and accession number are correct,

NO if at least one of them is incorrect,

? for an explanation, or

Q to have information for this specimen ignored."

?, print an explanation at the terminal and repeat the question.

"In telling SPECS about a new specimen, you gave a site, year, and accession number that have not been recorded in the site-yr-acc file. SPECS wants to know whether the site, year, and accession number are correct. If you say they are, SPECS will record them in the site-yr-acc file (and, of course, in the master-file record for the specimen). If you say the site, year, and accession number are not correct, SPECS will ask you for new information about the site, year, and accession number."

YES, remember that the site, year, and accession number are correct.

NO, ask for another site, year, and accession number by restarting step 5.13.2.4.6.

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

5.13.2.4.10.4 If the user said that the site, year, and accession number are correct, enter them in the site-yr-acc file and tell the user.

In the messages to be printed at the terminal, site, year and acc# are replaced by the current site, year, and accession number.

5.13.2.4.10.4.1 Write a new record in the site-yr-acc file.

5.13.2.4.10.4.2 Write messages at the terminal and on the session report to tell the user that the information has been entered. (SRE 98)

"Site site, year year, and accession number acc# are entered in the site-yr-acc file."

5.13.2.4.10.4.3 If there were already entries for the site, year, and accession number, write a session-report entry about them. (SRE 99)

5.13.2.4.11 If there is a "previous specimen," perform sequence checking for specimen numbers using the algorithm in Appendix C. If validation fails, print a message at the terminal and remember (but do not write) a session report entry about the specimen number. (SRE 100)

"The current specimen number does not immediately follow the last one given."

5.13.2.4.12 Validate the grid square information.

  5.13.2.4.12.1 Apply the algorithm in Appendix C.

  5.13.2.4.12.2 If the grid square or a coordinate is malformed, the grid-square information may be erroneous or of an unexpected type. Ask the user to supply the information again, broken into its component parts, then repeat the validation.

    5.13.2.4.12.2.1 Ask the user what kind of information was given.

      5.13.2.4.12.2.1.1 Print a question at the terminal.

      "What kind of grid square information did you give?"

      5.13.2.4.12.2.1.2 Wait for the user's response.

      5.13.2.4.12.2.1.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter GRID for a grid square,

POINT for a point provenience,

AREA for an area denoted by a range of coordinates,

NONE if you did not want to give grid square information,

OTHER if the information does not represent a grid square, point provenience, or area,

Q to have information about this specimen ignored, or

? for an explanation."

?, print an explanation at the terminal and repeat the question.

"SPECS thinks you are telling it about a new specimen, for which you gave grid square information. SPECS cannot interpret what you said, and is asking what kind of location the information described."

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information about this specimen."

GRID, remember that the grid-square type is G.

POINT, remember that the grid-square type is P.

AREA, remember that the grid-square type is A.

NONE, remember that the grid square type is N.

OTHER, remember that grid-square type is O.

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your response."

5.13.2.4.12.2.2 If the new grid-square type is not N, ask the user for the N-S coordinate of the point that best represents the grid square information.

5.13.2.4.12.2.2.1 Print a prompt at the terminal.

"Please enter the N - S coordinate of the point that best represents the grid square information:"

5.13.2.4.12.2.2.2 Wait for the user's response.

5.13.2.4.12.2.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have your answer to the question about the kind of grid square information ignored, or

a single integer representing the N - S coordinate of the point."

?, print an explanation at the terminal and repeat the prompt.

"In giving SPECS information about a new specimen, you gave it grid square information it could not understand. SPECS is asking you to repeat the information. It now wants the N - S coordinate (the number before the "L" or "R") of the single point on the site that best represents the grid square information you wanted to give."

Q, print messages at the terminal and repeat the question about grid square type by restarting step 5.13.2.4.12.2.1.

"At your request, SPECS is ignoring your answer to the question about grid square type."

an integer, assume it is the N - S coordinate.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response."

5.13.2.4.12.2.3 If the new grid-square type is not N, ask the user for the "L" or "R" of the point that best represents the grid square.

5.13.2.4.12.2.3.1 Print a prompt at the terminal.

"Please enter the L or R of the point that best represents the grid square information:"

5.13.2.4.12.2.3.2 Wait for the user's response.

5.13.2.4.12.2.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter L if the point is west of the site datum,
R if the site is east of the datum,
? for an explanation, or
Q to have your answer to the question about grid square type ignored."

?, print an explanation at the terminal and repeat the prompt.

"In telling SPECS about a new specimen, you gave it grid square information it could not understand. SPECS is asking you to repeat the information. It now wants you to enter the "L" or "R" from the coordinates of the point that best represents the grid square information you wanted to give."

L, remember that the point is in the left (west) half of the site.

R, remember that the point is in the right (east) half of the site.

Q, print messages at the terminal and repeat the question about grid square type by restarting step 5.13.2.4.12.2.1.

"At your request, SPECS is ignoring your answer to the question about grid square type."

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response."

5.13.2.4.12.2.4 If the new grid-square type is not N, ask the user for the E-W coordinate of the point that best represents the grid square information.

5.13.2.4.12.2.4.1 Print a prompt at the terminal.

"Please enter the E - W coordinate of the point that best represents the grid square information:"

5.13.2.4.12.2.4.2 Wait for the user's response.

5.13.2.4.12.2.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have your answer to the question about grid square type ignored, or

a single integer representing the E - W coordinate of the point."

?, print an explanation at the terminal and repeat the prompt.

"In telling SPECS about a new specimen, you gave it grid square information it could not understand. SPECS is asking you to repeat the information. It now wants the E - W coordinate (the number after the "L" or "R") of the single point on the site that best represents the grid square information you wanted to give."

Q, print a message at the terminal and repeat the question about grid-square type by restarting step 5.13.2.4.12.2.1.

"At your request, your answer to the question about grid square type is being ignored."

an integer, assume it is the E - W coordinate.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response."

5.13.2.4.12.2.5 Apply the validation algorithm in Appendix C, recognizing that the grid square information is already divided into N-S coordinate, L or R, and E-W coordinate and that the type of grid-square information is known.

5.13.2.4.12.3 If the user was not asked to repeat the grid square information, he does not know what kind of information SPECS thinks it was given. Print a message at the terminal to tell him.

In this message, _type_ is to be replaced by "area", "grid square", or "point provenience", as appropriate for the grid square type.

"The grid square information describes a _type_."

5.13.2.4.12.4 Remember (but do not write) a session report entry about the kind of grid square information.

5.13.2.4.12.4.1 If the grid-square type is N, remember the entry. (SRE 51)

5.13.2.4.12.4.2 If the grid-square type is A, remember the entry. (SRE 52)

5.13.2.4.12.4.3 If the grid-square type is O, remember the entry. (SRE 53)

5.13.2.4.12.4.4 If the grid-square type is G, remember the entry.
(SRE 54)

5.13.2.4.12.4.5 If the grid-square type is P, remember the entry. (SRE 55)

5.13.2.4.12.5 If validation for the current grid-square information showed that an E-W coordinate is negative, write a message at the terminal and remember (but do not write) one for the session report.
(SRE 56)

"The east-west coordinate is negative, and is probably an error."

5.13.2.4.12.6 If a real-number E-W coordinate was converted to an integer, print a message at the terminal and remember (but do not write) one for the session report. (SRE 57)

"The east-west coordinate was converted to an integer so that it could be represented."

5.13.2.4.12.7 If a real-number N - S coordinate was converted to an integer, print a message at the terminal and remember (but do not write) one for the session report. (SRE 58)

"The north-south coordinate was converted to an integer so that it could be represented."

5.13.2.4.12.8 If information about an area was converted to integers, print a message at the terminal and remember (but do not print) one for the session report. (SRE 59)

"The area whose coordinates you gave is represented by the point at its southeast corner."

5.13.2.4.12.9 If an E - W coordinate in the left half of the site is not within the expected range, print a message at the terminal and remember (but do not write) one for the session report. (SRE 60)

"The east-west coordinate is outside the range set by the validation constants for the left half of the site."

5.13.2.4.12.10 If an E - W coordinate in the right half of the site is not within the expected range, print a message at the terminal and remember (but do not write) one for the session report. (SRE 61)

"The east-west coordinate is outside the range set by the validation constants for the right half of the site."

5.13.2.4.12.11 If the N - S coordinate is not within the expected range, print a message at the terminal and remember (but do not write) one for the session report. (SRE 62)

"The north-south coordinate is outside the range set by the validation constants."

5.13.2.4.12.12 If the grid-square type is O, A, or P, the north-south and east-west coordinates do not represent all of the available grid-square information. Print a message at the terminal and remember (but do not write) one for the session report to tell the user that the full information will be added to the "location" field. (SRE 101)

"The information you originally provided for grid square will be added to information from the "location" input field."

5.13.2.4.13 Validate the unit information.

5.13.2.4.13.1 Apply the algorithm in Appendix C.

5.13.2.4.13.2 If validation fails, ask the user what kind of unit information he gave.

5.13.2.4.13.2.1 Print a question at the terminal.

"From what kind of unit was the specimen collected?"

5.13.2.4.13.2.2 Wait for the user's response.

5.13.2.4.13.2.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter NONE if there is no unit
information,

? for an explanation,

Q to request that information
from this specimen be
ignored, or

FEATURE,

BURIAL,

POSTHOLE or POST HOLE,

PIT or TEST PIT,

STRUCTURE, or

OTHER to indicate the type of
unit."

?, print an explanation at the terminal and repeat the question.

"You told SPECS about a new specimen. It cannot interpret what you said about the unit, and has asked you what kind of information you gave. Your answer to this question will determine what other questions SPECS will ask about the unit information."

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

FEATURE, BURIAL, POSTHOLE, POST HOLE, PIT, TEST PIT, STRUCTURE, NONE, or OTHER, consider it the unit type.

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

5.13.2.4.13.3 If the unit validation failed and the new unit type is not NONE or OTHER, ask the user to repeat the unit number.

5.13.2.4.13.3.1 Print a prompt at the terminal.

"Please enter the unit number:"

5.13.2.4.13.3.2 Wait for the user's response.

5.13.2.4.13.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to request that your answer to the question about the type of unit be ignored, or

a unit number."

?, print an explanation at the terminal and repeat the prompt.

"You told SPECS about a new specimen. It cannot interpret what you said about the unit, and wants you to repeat the unit number. A unit number may be a number or a number and a single letter. If you want to represent anything else, record "unit type" as "other" and SPECS will add any unit information you give it to the "location" field."

Q, print a message at the terminal and repeat the question about the unit type by returning to step 5.13.2.4.13.2.1.

"At your request, SPECS is ignoring your answer to the question about the kind of unit information."

anything else, assume that it is the unit number.

5.13.2.4.13.4 If the user was asked for information about the unit type and number, perform validation for the new information by repeating step 5.13.2.4.13, remembering that the unit type and number are known.

5.13.2.4.13.5 If the unit number included a letter, the unit information will be stored in two places: the numeric part of the information will be stored in the "unit" field, and the abbreviated unit information produced during validation will be added to the "location" field. In this case, print a message at the terminal and remember (but do not write) one for the session report. (SRE 102)

In the message to be printed at the terminal, <u>number</u> represents the numeric part of the unit identification and <u>short</u> represents the abbreviated information produced during validation.

"The unit information will be represented both as <u>number</u> in the "unit" field of the specimen-data record and as <u>short</u> in the "location" field."

5.13.2.4.13.6 If the unit type is "other", write a message at the terminal and remember (but do not write) one for the session report to tell the user that the unit information will be added to the "location" field. (SRE 103)

"The unit information for units of type "other" is added to the "location" field of the specimen-data record."

5.13.2.4.14 Validate the depth information.

5.13.2.4.14.1 Apply the algorithm in Appendix C.

5.13.2.4.14.2 If the surface information cannot be represented, print a message at the terminal and remember (but do not write) one for the session report. (SRE 41)

"SPECS cannot represent information about the depth of surface specimens. If you want SPECS to retain more information than that the specimen was from the surface, add the extra information to the "location" field."

5.13.2.4.14.3 If the level or zone number is malformed, print a message at the terminal and remember (but do not write) one for the session report. (SRE 42)

"SPECS does not recognize the information you have given as a level or zone number, but will treat it as one anyway."

5.13.2.4.14.4 If the depth measurement was truncated, print a message at the terminal and remember (but do not write) one for the session report. (SRE 43)

"The depth measurement is too long to be represented, and is truncated to three characters. You may represent the depth more fully by adding information to the "location" field."

5.13.2.4.14.5 If the level or zone number is too high, print a message at the terminal and remember (but do not print) one for the session report. (SRE 44)

"The level or zone number is greater than the maximum level or zone number allowed by the validation constants."

5.13.2.4.14.6 If the depth type is "other," write a message at the terminal, and remember (but do not write) one for the session report, to tell the user that the depth information will be added to the "location" field. (SRE 104)

"SPECS cannot represent the kind of depth information you have given, and has encoded it as "other." The depth information you gave will be added to the information from the "location" input field."

5.13.2.4.15 If there is a "previous specimen," use the algorithm in Appendix C to perform sequence checking for depth information. If validation fails, print a message at the terminal and remember (but do not write) one for the session report.

(SRE 105)

"The depth information from this location is not recorded in order of increasing depth. Please check the location information."

5.13.2.4.16 Validate the weight information.

5.13.2.4.16.1 Apply the algorithm in Appendix C.

5.13.2.4.16.2 If validation fails, print a message at the terminal.

"SPECS cannot recognize the "weight" information you gave as a weight in ounces."

5.13.2.4.16.3 If the validation failed, ask the user for new weight information, then perform validation for it.

5.13.2.4.16.3.1 "Please enter the weight, in ounces:"

5.13.2.4.16.3.2 Wait for the user's response.

5.13.2.4.16.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to request that information from this specimen be ignored,

NONE if there is no weight information, or

a weight in ounces."

?, print an explanation at the terminal and repeat the prompt.

"You gave SPECS information about a new specimen. It cannot interpret the weight information you gave, and has asked you to repeat it."

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else (including NONE), assume that it is a weight.

5.13.2.4.16.3.4 Perform validation for the new weight by restarting step 5.13.2.4.16.

5.13.2.4.16.4 If the weight was rounded, print a message at the terminal and remember (but do not print) one for the session report. (SRE 73)

"The weight you gave was rounded to the nearest 0.1 ounce."

5.13.2.4.16.5 If the weight was treated as "> 3276.7", print a message at the terminal and remember (but do not print) one for the session report. (SRE 74)

"The weight you gave was too large to represent, and is being recorded as greater than 3276.7 oz (almost 205 lbs). This weight is probably erroneous."

5.13.2.4.17 Validate the count information.

5.13.2.4.17.1 Apply the algorithm in Appendix C.

5.13.2.4.17.2 If validation fails, get new count information and validate it.

5.13.2.4.17.2.1 Print a message at the terminal.

"SPECS cannot interpret the count information."

5.13.2.4.17.2.2 Ask the user to identify the type of count information.

5.13.2.4.17.2.2.1 Print a question at the terminal.

"What kind of count information did you give?"

5.13.2.4.17.2.2.2 Wait for the user's response.

5.13.2.4.17.2.2.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter BAG,

JAR,

BOX,

VIAL,

PIECES,

OTHER, or

MIXED to indicate the kind of item counted, or

NONE if there is no count information,

? for an explanation, or

Q to request that SPECS ignore information about this specimen."

?, print an explanation at the terminal and repeat the question.

"You told SPECS about a new specimen. It cannot interpret the count information, and is asking what kind of units were being counted. Your answer to this question will determine what other questions SPECS will ask about the count information."

BAG, JAR, BOX, VIAL, OTHER, PIECES, MIXED, or NONE, consider it the count type.

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information for this specimen."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

5.13.2.4.17.2.3 If the new count type is not "NONE," ask the user to repeat the count.

5.13.2.4.17.2.3.1 Print a prompt at the terminal.

"Please enter the count:"

5.13.2.4.17.2.3.2 Wait for the user's response.

5.13.2.4.17.2.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to request that your answer to the question about the type of count information be ignored,

or

the correct count."

?, print an explanation at the terminal and repeat the prompt.

"You told SPECS about a new specimen. It cannot interpret the count information, and is asking you to repeat the number of items counted."

Q, print a message at the terminal and go back to the question about count type by restarting step 5.13.2.4.17.2.2.

"At your request, SPECS is ignoring your answer to the question about the kind of count information."

anything else, assume it is the new count.

5.13.2.4.17.2.4 Validate the new count information by restarting step 5.13.2.4.17, remembering that the count type and count are known.

5.13.2.4.17.3 If the count is negative, print a message at the terminal and remember (but do not write) one for the session report. (SRE 36)

"The count is negative, and is probably an error."

5.13.2.4.17.4 If the count was rounded to produce an integer, print a message at the terminal and remember (but do not write) one for the session report. (SRE 37)

"The count was rounded to produce an integer."

5.13.2.4.17.5 If the count was recoded as -32,768, print a message at the terminal and remember (but do not write) one for the session report. (SRE 38)

"The count was too far negative to be represented, and has been changed to -32,768."

5.13.2.4.17.6 If the count was recoded as 32,767, print a message at the terminal and remember (but do not write) one for the session report. (SRE 39)

"The count was too large to be represented, and has been changed to 32,767."

5.13.2.4.17.7 If the container count was too large, print a message at the terminal and remember (but do not write) one for the session report. (SRE 40)

"The number of boxes, bags, vials, or jars is larger than the maximum number of containers allowed by the validation constants."

5.13.2.4.17.8 If the count type is "other," print a message at the terminal to suggest that the user may want to put more information in the "description" field.

"The count type is recorded as "other." If you want more detailed information to be stored, you may add it to the "description" field."

5.13.2.4.18 Validate the location information.

5.13.2.4.18.1 Apply the algorithm in Appendix C.

5.13.2.4.18.2 If the information was truncated, print a message at the terminal and remember (but do not write) one for the session report. (SRE 63)

"The location information was truncated to 127 characters."

5.13.2.4.19 Validate the description information.

5.13.2.4.19.1 Apply the algorithm in Appendix C.

5.13.2.4.19.2 If "disc" or "discs" was changed to "disk" or "disks", print a message at the terminal and remember (but do not write) one for on the session report. (SRE 45)

""disc" was changed to "disk" to standardize spelling."

5.13.2.4.19.3 If "C.S.P.P.", "C. S. P. P", or "CS PP" was changed to CSPP, print a message at the terminal and remember (but do not write) one for the session report. (SRE 46)

In the message, <u>old</u> <u>spelling</u> is replaced by the spelling that was changed.

"<u>old</u> <u>spelling</u>" was changed to "CSPP" to standardize spelling.

5.13.2.4.19.4 If the description was truncated, print a message at the terminal and remember (but do not write) one for the session report. (SRE 47)

"The description was truncated to 127 characters so that it could be represented."

5.13.2.4.20 Use the algorithm for type-description validation (Appendix C) to check the consistency of the description and specimen type. If the validation fails, print a message at the terminal and remember (but do not write) one for the session report. (SRE 106)

In the message, word is replaced by the words found in the "words from descriptions" column of table 10 (in Appendix C), and type is the corresponding specimen type from the same table.

"Most specimens whose descriptions contain "word" are of type type, but this one is not. Please check the specimen type and description."

5.13.2.4.21 Perform kind-count-weight validation to check the consistency of the specimen type, description, count, and weight.

In the messages printed at the terminal during this step, kind is replaced by the specimen kind, as given in the "kind" column of Table 11 (in Appendix C).

5.13.2.4.21.1 Apply the algorithm in Appendix C.

5.13.2.4.21.2 If validation fails because soil samples or washings are too heavy, write a message at the terminal and remember (but do not write) one for the session report. (SRE 107)

"The specimen weight is greater than the maximum given in the validation constants for <u>kind</u>."

5.13.2.4.21.3 If validation fails because the count is too large, write a message at the terminal and remember (but do not write) one for the session report. (SRE 108)

"The number of pieces in the specimen is larger than is allowed by the validation constants for <u>kind</u>."

5.13.2.4.21.4 If validation fails because the density is not within the expected range, write a message at the terminal and remember (but do not write) one for the session report. (SRE 109)

"The number of pieces per ounce is not within the range given by the validation constants for <u>kind</u>."

5.13.2.4.22 Write the information about the new specimen at the terminal, showing the input fields and the way SPECS encodes them. (Terminal Report 5).

5.13.2.4.23 Ask the user whether the information he was shown is correct.

5.13.2.4.23.1 Print a prompt at the terminal.

"Is the information you were just shown correct?"

5.13.2.4.23.2 Wait for the user's response.

5.13.2.4.23.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter YES if all of the information is correct,

NO if some of the information is incorrect,

SHOW to have the information repeated,

Q to have SPECS ignore the information from this specimen, or

? for an explanation."

?, print an explanation at the terminal and repeat the prompt.

"SPECS is ready to write an insert-data record to add the information you were shown to the master file during the next batch update. SPECS is asking whether you want a final chance to change the information before the insert-data record is written. If you say the information is incorrect, SPECS will let you change the information, and will use any new data that you give in the insert-data record. If you say that the information you were shown is correct, you may use a CHANGE command to change it later."

YES, remember that no information needs changing.

NO, remember that the information is incorrect.

SHOW, print Terminal Report 5 and repeat the question.

Q, print a message at the terminal and stop processing information for this specimen.

"At your request, SPECS will ignore the information you gave it about this specimen."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

5.13.2.4.24 If the information is incorrect, allow the user to change the information until he says he does not want to change more.

5.13.2.4.24.1 Ask the user which input field he wants to change.

5.13.2.4.24.1.1 Print a question at the terminal.

"For which field do you want to change information?"

5.13.2.4.24.1.2 Wait for the user's response.

5.13.2.4.24.1.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter ? for an explanation,

Q if you do not want to change information,

SHOW to see the current information again, or

COUNT,

DEPTH,

DESCRIPTION,

GRID,

LOCATION,

SITE,

SPECIMEN,

UNIT,

WEIGHT, or

YEAR."

?, print an explanation at the terminal and repeat the question.

"You told SPECS to change information about this specimen before it writes the information in the transaction file. SPECS wants to know in which input field the information you want to change was entered."

Q, assume that the information for the specimen is correct, print a message at the terminal, and stop executing step 5.13.2.4.24.

"At your request, SPECS will no longer expect you to change information about the current specimen."

SHOW, print Terminal Report 5 and repeat the question.

a field name (COUNT, DEPTH, DESCRIPTION, GRID, LOCATION, SITE, SPECIMEN, UNIT, WEIGHT, or YEAR), consider it the name of the field to be changed.

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your response."

5.13.2.4.24.2 Ask the user for a new value for the field to be changed.

In the following, <u>field</u> is replaced by "count," "depth," "description," "grid," "location," "site," "specimen number," "unit," "weight," or "year."

5.13.2.4.24.2.1 Print a prompt at the terminal.

"Please enter new information about <u>field</u>:"

5.13.2.4.24.2.2 Wait for the user's response.

5.13.2.4.24.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have SPECS ignore your answer to the question about the field to be changed,

: to have this menu repeated,

NONE if there is no new value, or

a new value for the input field <u>field</u>."

?, print an explanation at the terminal and repeat the prompt.

"You told SPECS to change information from the <u>field</u> input field for the current specimen. SPECS wants to know what new information you want used to replace the old value of the field."

Q, print a message at the terminal and repeat the question about the input field to be changed by restarting step 5.13.2.4.24.

"At your request, SPECS is ignoring your answer to the question about the field to be changed."

anything else (including NONE), assume it is the new value for the input field.

5.13.2.4.24.3 If the new information is not the same as the current value of the field, repeat the single-field validation for the input field. The validation algorithms, and the consequences of failure, are those that would be used if the new value was given as part of a CHANGE command (step 5.7), except that:

1. Session report entries are remembered, but not written.
2. A "Q" command causes the input field, and the specimen-data fields derived from it, to revert to their values at the beginning of step

5.13.2.4.24. The messages printed to acknowledge the "Q" command are changed accordingly.

3. Responses that would have caused the CHANGE command to be ignored are handled by printing an error message at the terminal, then asking for a new value by restarting step 5.13.2.4.24.2.

5.13.2.4.24.4 If the single-field validation in step 5.13.2.4.24.3 was performed, forget all unwritten session report entries that were remembered from single-field validation of earlier values of the input field. If the input field was copied from the previous specimen, also forget the unwritten session report entry that told the user so.

5.13.2.4.24.5 If the specimen number was changed, perform sequence checking for specimen numbers, using the same logic as for the original number for this specimen (step 5.13.2.4.11). Forget any unwritten session report entries generated during the earlier sequence checking.

5.13.2.4.24.6 If the site, year, or accession number changed, perform site-yr-acc# validation. The validation algorithms, and consequences of failure, are those that were used when site-yr-acc# validation was performed for the original values given for this specimen (steps 5.13.2.4.9.1 to 5.13.2.4.10), except that:

1. If validation fails, the site, year, and accession number revert to their values at the beginning of step 5.13.2.4.24.
2. A "Q" command causes the input field that was changed, and the specimen-data fields derived from it, to revert to their values at the beginning of step 5.13.2.4.24. The messages printed to acknowledge the "Q" command are changed accordingly.

5.13.2.4.24.7 If site, year, accession number, grid square, unit, or depth was changed, perform sequence checking for depths. Use the same logic as for the original information for this specimen (step 5.13.2.4.15), and forget any unwritten session report entries generated during the earlier sequence checking for depths.

5.13.2.4.24.8 If specimen type or description changed, perform type-description validation. Use the same logic as for the original type and description for this specimen (step 5.13.2.4.20), and forget any unwritten session report entries generated during the earlier type-description validation.

5.13.2.4.24.9 If specimen type, description, count, or weight changed, perform kind-count-weight validation. Use the same logic as for the original type, description, count, and weight for this specimen (step 5.13.2.4.21), and forget any unwritten session report entries generated during the earlier kind-count-weight validation.

5.13.2.4.24.10 If the unit information, grid square information, or depth information was changed, forget any old information that was to be automatically added to the "location" field.

5.13.2.4.24.11 Write messages at the terminal to tell the user about the new value.

5.13.2.4.24.11.1 If the field to be changed was "count", tell the user.

In this message, <u>count</u> is replaced by the count information given by the user, <u>type</u> is replaced by the count type from validation, and <u>number</u> is replaced by the count from validation.

"The new count information is <u>count</u>, which is encoded:

Count type: <u>count type</u>

Count: <u>number</u>"

5.13.2.4.24.11.2 If the field to be changed was "depth," tell the user.

In this message, <u>depth</u> is replaced by the depth information given by the user, <u>type</u> is replaced by the depth type from validation, and <u>depth ID</u> is replaced by the depth from validation.

5.13.2.4.24.11.2.1 Tell the user about the information that is not in the "location" field.

"The new depth information is depth, which is encoded:

Depth type: type

Depth: depth ID"

5.13.2.4.24.11.2.2 If the depth information was added to the "location" field, tell the user.

"Depth was added to the "location" field."

5.13.2.4.24.11.3 If the field to be changed was "description," tell the user.

In this message, description is replaced by the description information given by the user, and length is replaced by the description length from validation.

"The new description information is <u>description</u>.
The description length is <u>length</u>."

5.13.2.4.24.11.4 If the field to be changed was "grid square," tell the user.

In this message, <u>grid square</u> is replaced by the grid square information given by the user, <u>type</u> is replaced by the grid square type from validation, <u>N-S coord</u> is replaced by the N-S coordinate from validation, <u>L-R half</u> is replaced by the L-R half from validation, and <u>E-W coord</u> is replaced by the E-W coordinate from validation.

5.13.2.4.24.11.4.1 Tell the user about the information that is not in the "location" field.

"The new grid square information is <u>grid square</u>, which is encoded:

Grid square type: <u>type</u>

N - S coord: <u>N-S coord</u>

L - R half: <u>L-R half</u>

E - W coord: <u>E-W coord</u>"

5.13.2.4.24.11.4.2 If the grid square information was added to the "location" field, tell the user.

"<u>Grid square</u> was added to the "location" field."

5.13.2.4.24.11.5 If the field to be changed was "location," tell the user.

In this message, <u>location</u> is replaced by the location information given by the user, and <u>length</u> is replaced by the location length from validation.

"The new location information is <u>location</u>.
The location length is <u>length</u>."

5.13.2.4.24.11.6 If the field to be changed was "site," tell the user.

In this message, <u>site</u> is replaced by the site information given by the user.

"The new site information is <u>site</u>."

5.13.2.4.24.11.7 If the field to be changed was "unit," tell the user.

In this message, <u>unit</u> is replaced by the unit information given by the user, <u>type</u> is replaced by the unit type from validation, <u>added</u> is replaced by any unit information to be added to the "location" field, and <u>unit</u> <u>ID</u> is replaced by the unit from validation.

5.13.2.4.24.11.7.1 Tell the user about the information that is not in the "location" field.

"The new unit information is <u>unit</u>, which is encoded:

Unit type: <u>type</u>

Unit: <u>unit</u> <u>ID</u>"

5.13.2.4.24.11.7.2 If unit information was added to the "location" field, tell the user.

"Added was added to the "location" field."

5.13.2.4.24.11.8 If the field to be changed was "weight", tell the user.

In this message, weight is replaced by the weight information given by the user, type is replaced by the weight type from validation, and ounces is replaced by the weight from validation.

"The new weight information is weight, which is encoded:

Weight type: weight type

Weight: ounces"

5.13.2.4.24.11.9 If the field to be changed was "year," tell the user.

In this message, year is replaced by the year information given by the user.

"The new year information is year."

5.13.2.4.24.11.10 If the field to be changed was "specimen number," tell the user.

In this message, <u>spec#</u> is replaced by the specimen number given by the user, and <u>acc#</u>, <u>type</u>, and <u>seq#</u> are replaced by the accession number, specimen type, and sequence and continuation number from validation.

"The new specimen number is <u>spec#</u>, which is encoded:

Accession number: <u>acc#</u>

Specimen type: <u>type</u>

Sequence and continuation number: <u>seq#</u>"

5.13.2.4.24.12 Restart step 5.13.2.4.24 to allow the user to change more information.

5.13.2.4.25 Write a session report entry summarizing the final values of the input fields in the data for the specimen.
(SRE 110)

5.13.2.4.26 Write the remembered, but unwritten, session report entries for the specimen.

5.13.2.4.27 Add any information that must be automatically added to the location field. Concatenate a period followed by a blank to the beginning of the information from each field to be added, then concatenate the fields to be added to the end of the "location" field, in whatever order is convenient.

5.13.2.4.28 If information was added to the "location" field, use the algorithm in Appendix C to validate the new value of "location." If the information was truncated, print messages at the terminal and on the session report. (SRE 63)

"The location information was truncated to 127 characters."

5.13.2.4.29 Write an insert-data record on the transaction file, using these values:

Accession number, sequence and continuation number, and specimen type: from validation of the specimen number.

Transaction number: the next transaction number to be assigned.

Transaction type code: I.

Site: from validation of the site information, or from the previous specimen.

Year: from validation of the year information, or from the previous specimen.

Unit type and unit: from validation of the unit information.

Grid-square type, N-S coord, L-R half, E-W coord: from validation of the grid square information.

Depth type, depth: from validation of the depth information.

Count type, number: from validation of the count information.

Weight type, weight: from validation of the weight information.

Location length: from validation of the location information.

Description length: from validation of the description information.

Location: from processing the unit, grid square, and depth information, and from validation of the location information.

Description: from validation of the description information.

5.13.2.4.30 Increment the next transaction number to be assigned.

5.13.2.4.31 Write a message at the terminal about the insert-data record.

In this message, trans# is replaced by the transaction number of the insert-data record.

5.13.2.4.31.1 "This information about a new specimen will be added to the master file during the next batch update:"

5.13.2.4.31.2 Write Terminal Report 5 at the terminal.

5.13.2.4.31.3 "(Transaction # trans#)"

5.13.2.4.32 Write a session report entry about the insert-data record. (SRE 111)

5.13.2.4.33 Consider the current specimen to be the new "previous specimen."

5.13.2.4.34 Restart step 5.13.2 to get information about another specimen.

5.14 If the response to the request for a command was:

RESET, allow the user to reset a password or validation constant, and repeat the request for a command.

5.14.1 Find out what to reset.

5.14.1.1 If the rest of the command is an entry (other than ? and Q) in the menu below, or the name of a validation constant or password listed in Appendix B, it indicates the value to be reset.

5.14.1.2 Otherwise, ask the user what to reset.

5.14.1.2.1 Print a question at the terminal.

"What information do you want to reset?"

5.14.1.2.2 Wait for the user's response.

5.14.1.2.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter ? for an explanation,

Q to request that the RESET command be ignored,

CANCEL to reset the cancel password,

CHANGE to reset the change password,

CONTROL to reset the validation-control password,

DELETE to reset the delete password,

INPUT to reset the input password,

MASTER to reset the master password,

SET-ACC to reset the set-acc password,

SHOW-ACC to reset the show-acc password,

UPDATE to reset the update password,

A-COUNT to rest the type-"a" count,

B-COUNT to reset the type-"b" count,

BEAD COUNT to reset the bead count,

E-COUNT to reset the type-"e" count,

FLAKE COUNT to reset the flake count,

H-COUNT to reset the type-"h" count,

M-COUNT to reset the type-"m" count,

P-COUNT to reset the type-"p" count,

MAX-A to reset the maximum type-"a" density,

MAX-B to reset the maximum type-"b" density,

MAX BEAD to reset the maximum bead density,

MAX-E to reset the maximum type-"e" density, MAX FLAKE to reset the maximum flake density,

MAX-H to reset the maximum type-"h" density,

MAX-M to reset the maximum type-"m" density,

MAX-P to reset the maximum type-"p" density,

MIN-A to reset the minimum type-"a" density,

MIN-B to reset the minimum type-"b" density,

MIN BEAD to reset the minimum bead density,

MIN-E to reset the minimum type-"e" density,

MIN FLAKE to reset the minimum flake density,

MIN-H to reset the minimum type-"h" density, MIN-M to reset the minimum type-"m" density,

MIN-P to reset the minimum type-"p" density,

CONTAINERS to reset the container count,

SOIL to reset the soil-sample weight,

WASHINGS to reset the washings weight,

MAX LEVEL to reset the maximum level,

MAX E-W FOR L to reset the maximum E-W for "L",

MIN E-W FOR L to reset the minimum E-W for "L",

MAX E-W FOR R to reset the maximum E-W for "R",

MIN E-W FOR R to reset the minimum E-W for "R", MAX N-S to reset the maximum N-S,

or

MIN N-S to reset the minimum N-S."

?, print an explanation at the terminal and repeat the question.

"You have told SPECS to reset a password or validation constant. SPECS wants to know the name of the password or constant to be changed. If you are authorized to change the password or constant, SPECS will print its current value, then ask you for a new one."

Q, print a message at the terminal and stop executing the RESET command.

"At your request, SPECS is ignoring the RESET command."

any other selection from the menu, or the name of a password or validation constant in the system parameters file, remember to reset the appropriate password or constant.

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your response."

5.14.2 Print a session report entry to summarize the information in the command. (SRE 112)

5.14.3 Passwords may be reset only by a user who knows the master password, and validation constants only by a user who knows the validation-control password. If the password needed for the value to be reset has not been given earlier in the session, ask the user for it.

In this step, <u>password</u> is replaced by the name of the required password ("master" or "validation-control"). <u>Kinds</u> is replaced by "passwords" if the value being replaced is a password, or by "validation constants" if the value is a validation constant.

5.14.3.1 "Please enter the <u>password</u> password:"

5.14.3.2 Wait for the user's response.

5.14.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter Q have the RESET command ignored,

? for an explanation, or

the correct password."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS that you want to reset a <u>kind</u>. SPECS wants you to tell it the <u>password</u> password to demonstrate that you are authorized to reset the information."

Q, print messages at the terminal and on the session report, and stop trying to get the password. (SRE 113)

"At your request, SPECS is ignoring your RESET command."

the correct password, remember that it was given.

anything else (the first 2 times an incorrect password is given), print a message at the terminal and repeat the prompt.

"SPECS cannot recognize your response as the correct password."

anything else (the third time), print messages at the terminal and on the session report, and stop trying to get the master password. (SRE 113)

"SPECS did not receive the correct password, and cannot allow you to reset a kind."

5.14.4 Print a message at the terminal to tell the user about the current value of the password or constant to be reset.

In the message to be printed at the terminal, name is replaced by the name of the password or constant (as given in the menu entry, following the "to reset"), and value is replaced by its current value.

"The current value of name is value."

5.14.5 If the value to be reset is a validation constant, print a session report entry about its old value. (SRE 114)

5.14.6 If the value to be reset is a password, ask the user for its new value.

In this step, name is replaced by the name of the password to be reset, and function is replaced by its function, as given in Table 13.

5.14.6.1 Print a prompt at the terminal.

"Please enter the new value for the <u>name</u>:"

5.14.6.2 Wait for the user's response.

5.14.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

: to have this menu repeated,

Q to have the RESET command ignored,

SAME to leave the <u>name</u> unchanged, or

the new <u>name</u>."

?, print an explanation at the terminal and repeat the prompt. "You have told SPECS to reset the <u>name</u>, which controls <u>function</u>. SPECS wants you to give it a new value, 1 to 8 characters long, for the password."

Q or SAME, print messages at the terminal and on the session report, and stop executing the RESET command. (SRE 115)

"At your request, the value of <u>name</u> remains unchanged."

any other series of 1 to 8 characters, consider it the new password.

anything else, print a mesage at the terminal and repeat the prompt.

"All passwords must have 1 to 8 characters."

5.14.7 If the value to be reset is a validation constant, ask the user for its new value.

In the message to be printed at the terminal, <u>name</u> is replaced by the name of the constant to be reset, and <u>function</u> is replaced by its function, as given in Table 14.

5.14.7.1 Print a prompt at the terminal.

"Please enter the new value for the <u>name</u>:"

5.14.7.2 Wait for the user's response.

5.14.7.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have the RESET command ignored,

SAME to leave the <u>name</u> unchanged, or

the new <u>name</u>."

?, print an explanation at the terminal and repeat the prompt.

"You have told SPECS you want to reset the <u>name</u>, which represents <u>function</u>. SPECS wants you to give it a new value for the validation constant."

Q or SAME, print messages at the terminal and on the session report, and stop executing the RESET command. (SRE 114)

"At your request, the value of <u>name</u> remains unchanged."

any value with the characteristics shown in Appendix B for the validation constant, consider it the new value for the constant.

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your answer."

5.14.8 Rewrite the record with the changed value in the system parameters file.

5.14.9 Write a session report entry about the new value.

5.14.9.1 If a password was changed, write a session report entry that does not give the new value. (The user can get the passwords printed on the session report later, if he wants.) (SRE 116)

5.14.9.2 If a validation constant was changed, write its new value on the session report. (SRE 117)

5.15 If the response to the request for a command was:

SET-ACC, change or delete one or more records in the site-yr-acc file, then repeat the request for a command.

5.15.1 If there is no more information in the command, the user must be asked for the site, year, and accession number.

5.15.2 If more information is given in the command, analyze it.

5.15.2.1 If the information has a left parenthesis and, later, a right parenthesis, assume that anything before the left parenthesis is information about a site, anything between the parentheses is a year, and anything after the right parenthesis is information about an accession number.

5.15.2.2 If the information does not have the parentheses, the user must be asked for the site, year, and accession number.

5.15.2.3 If information about the site was found, analyze it.

5.15.2.3.1 If the site information begins with "FOR" followed by at least one blank, remove the "FOR" from the site information.

5.15.2.3.2 Remove any blanks at the beginning and end of the site information.

5.15.2.3.3 Use the algorithm in Appendix C to validate the site information. If validation fails, the user must be asked for the site.

5.15.2.4 If the year was found, use the algorithm in Appendix C to validate it.

5.15.2.4.1 Apply the algorithm.

5.15.2.4.2 If the year was missing, the user must be asked for it.

5.15.2.4.3 If the year is too early, print messages at the terminal and on the session report. (SRE 77)

"The year you have given is too early to be correct for any of the current RLA collections, and is probably erroneous."

5.15.2.4.4 If the year is too late, print messages at the terminal and on the session report. (SRE 78)

"The year you have given has not yet arrived."

5.15.2.5 If information about the accession number was found, analyze it.

5.15.2.5.1 If the information begins with "TO" followed by one or more blanks, remove the "TO" from the accession number.

5.15.2.5.2 Ignore any blanks at the beginning of the accession number.

5.15.2.5.3 If the resulting accession number information has no characters, the user must be asked for the accession number.

5.15.2.5.4 If the accession number is not NONE, bur has at least one character, use the algorithm in Appendix C to validate it. If validation fails, the user must be asked for the accession number.

5.15.2.5.5 If the accession number information is NONE, remember to delete the records for the site and year from the site-yr-acc file.

5.15.3 If earlier processing shows that the user must be asked for the site, ask him for it.

5.15.3.1 Print a prompt at the terminal.

"Please enter the site identification:"

5.15.3.2 Wait for the user's response.

5.15.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to request that the SET-ACC command be ignored, or

a site identification."

?, print an explanation at the terminal and repeat the prompt.

"You have asked SPECS to set the accession number for a specified site and year. SPECS could not find the site identification in your command, and is asking you to repeat the information."

Q, print a message at the terminal and stop executing the SET-ACC command.

"At your request, SPECS is ignoring the SET-ACC command."

anything else, assume it is the site identification.

5.15.3.4 Use the algorithm in Appendix C to validate the assumed site identification. If validation fails, print a message at the terminal and get new site information by restarting step 5.15.3.

"SPECS cannot interpret your response as a site identification."

5.15.4 If earlier processing shows that the user must be asked for the year, ask him for it.

5.15.4.1 Print a prompt at the terminal.

"Please enter the year:"

5.15.4.2 Wait for the user's response.

5.15.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to request that the SET-ACC command be ignored, or

a year."

?, print an explanation at the terminal and repeat the prompt.

"You have asked SPECS to set the accession number for a specified site and year. SPECS could not find the year in your command, and is asking you to repeat the information."

Q, print a message at the terminal and stop executing the SET-ACC command.

"At your request, SPECS is ignoring the SET-ACC command."

anything else, assume it is the year.

5.15.4.4 Use the algorithm in Appendix C to validate the assumed year.

5.15.4.4.1 If the year information is missing or malformed, print a message at the terminal and get another year by restarting step 5.15.4.

"SPECS cannot interpret your response as a year."

5.15.4.4.2 If the year is too early, print messages at the terminal and on the session report.

"The year you have given is too early to be correct for any of the current RLA collections, and is probably erroneous."

5.15.4.4.3 If the year is too late, print messages at the terminal and on the session report.

"The year , you have given has not yet arrived."

5.15.5 If earlier processing showed that the user must be asked for the accession number, ask him for it.

5.15.5.1 Print a prompt at the terminal.

"Please enter the accession number:"

5.15.5.2 Wait for the user's response.

5.15.5.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to reqest that the SET-ACC command be ignored,

NONE if you wish to delete the site and year from the site-yr-acc file, or

an accession number."

?, print an explanation at the terminal and repeat the prompt.

"You have asked SPECS to set the accession numbr for a specified site and year. SPECS could not find the new accession number in the command, and is asking you to repeat the information."

NONE, remember to delete the records for the site and year.

Q, print a message at the terminal and stop executing the SET-ACC command.

"At your request, SPECS is ignoring the SET-ACC command."

anything else, assume it is an accession number.

5.15.5.4 If an accession number was given, use the algorithm in Appendix C to validate it. If validation fails, print a message at the terminal and get another accession number by restarting step 5.15.5.

"SPECS cannot interpret your response as an accession number."

5.15.6 Print a session report entry to summarize the information in the command. (SRE 118)

5.15.7 Concatenate the year to the end of the site designation to create a key for the site-yr-acc file, then read all of the records in the file that have the key.

5.15.8 Print messages at the terminal and on the session report to tell the user about any records that were read.

In the message to be printed at the terminal, site is replaced by the site, year is replaced by the year, and acc# is replaced by the accession number to be recorded for them. Old acc# is replaced by a list of the accession numbers that were in the records read from the file.

5.15.8.1 If no records were read, tell the user. (SRE 119)

"No accession number was known for site (year)."

5.15.8.2 If records were read, tell the user about their accession numbers. (SRE 99)

"The accession number(s) for site (year) were already recorded as old acc#."

5.15.9 If the file already had records with the key and the user gave a new accession number, ask him whether the new accession number replaces the ones already in the file.

5.15.9.1 Print a question at the terminal.

"Do you want the new accession number to replace the old ones?"

5.15.9.2 Wait for the user's response.

5.15.9.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter YES if the old accession numbers are incorrect,

NO if they are correct,

? for an explanation, or

Q to request that the SET-ACC command be ignored."

?, print an explanation at the terminal and repeat the question.

"You have asked SPECS to reset the accession number for a specified site and year. SPECS found that at least one other accession number was already recorded for that site and year, and wonders whether the old accession numbers should be replaced by the new accession number you gave. If you say that the new accession number replaces the old ones, SPECS will delete the old numbers from the site-yr-acc file. Otherwise, the site-yr-acc file will have entries for both the new accession number and the old ones, and SPECS will consider all of the accession numbers correct."

YES, remember to delete the records for the site and year.

NO, do nothing.

Q, print messages at the terminal and on the session report, then stop executing the SET-ACC command. (SRE 120)

"At your request, SPECS is ignoring the SET-ACC command."

anything else, print a message at the terminal and repeat the prompt.

"SPECS cannot interpret your response."

5.15.10 If the records for the site and year are to be deleted, print messages at the terminal and on the session report, and delete all of the appropriate site-yr-acc records. (SRE 121)

In the message printed at the terminal, old acc# is replaced by a list of the accession numbers from the records that are being deleted, site is replaced by the site, and year is replaced by the year.

"Accession numbers old acc# are no longer recorded for site (year)."

5.15.11 If a new accession number was given, write a site-yr-acc record with it and the key generated from the site and year, then write messages at the terminal and on the session report. (SRE 98)

In the message printed at the terminal, acc# is replaced by the accession number from the record that is being added, site is replaced by the site, and year is replaced by the year.

"Accession number acc# is now recorded for site (year)."

5.16 If the response to the request for a command was:

SHOW-ACC, read the accession number for a specified site and year from the site-yr-acc file, show it to the user, and repeat the request for a command.

5.16.1 If there is no more information in the command, the user must be asked for the site and year.

5.16.2 If more information was given in the command, analyze it.

5.16.2.1 If the information has a left parenthesis and, later, a right parenthesis, assume that anything before the left parenthesis is a site, anything between the parentheses is a year, and anything after the right parenthesis is to be ignored.

5.16.2.2 If the information does not have the parentheses, the user must be asked for the site and year.

5.16.2.3 If information is being ignored, print a message at the terminal.

In the message, <u>extra</u> is replaced by the information that is being ignored.

"Extra information <u>extra</u> is being ignored."

5.16.2.4 If the site information was found, analyze it.

5.16.2.4.1 If the site information begins with "FOR" followed by at least one blank, remove the "FOR" from the site information.

5.16.2.4.2 Remove any blanks at the beginning and end of the site information.

5.16.2.4.3 Use the algorithm in Appendix C to validate the site information. If validation fails, remember to ask the user for the site.

5.16.2.5 If the year was found, use the algorithm in Appendix C to validate it. If the year was missing or malformed, the user must be asked for it.

5.16.3 If previous processing shows that the user must be asked for the site, ask him for it.

5.16.3.1 Print a prompt at the terminal.

"Please enter the site identification:"

5.16.3.2 Wait for the user's response.

5.16.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to request that the SHOW-ACC command be ignored, or

a site identification."

?, print an explanation at the terminal and repeat the prompt.

"You have asked SPECS to show the accession number for a specified site and year. SPECS could not find the site identification in your command, and is asking you to repeat the information."

Q, print a message at the terminal and stop executing the SHOW-ACC command.

"At your request, SPECS is ignoring the SHOW-ACC command."

anything else, assume it is the site identification.

5.16.3.4 Use the algorithm in Appendix C to validate the assumed site information. If validation fails, print a message at the terminal and get new site information by restarting step 5.16.3.

"SPECS cannot interpret your response as a site identification."

5.16.4 If earlier processing shows that the user must be asked for the year, ask him for it.

5.16.4.1 Print a prompt at the terminal.

"Please enter the year:"

5.16.4.2 Wait for the user's response.

5.16.4.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

5.16.5 Print a session report entry summarizing the information in the command. (SRE 122)

5.16.6 Concatenate the year to the end of the site information to create a key for the site-yr-acc file, then read all of the records in the file that have the key.

5.16.7 Write messages at the terminal and on the session report to tell the user about the records that were read.

In the messages printed at the terminal, <u>acc#</u> is replaced by a list of the accession numbers from the records that were read, <u>site</u> is replaced by the site, and <u>year</u> is replaced by the year.

5.16.7.1 If no records were read, tell the user. (SRE 123)

"No accession numbers are recorded for <u>site</u> (<u>year</u>).

5.16.7.2 If records were read, tell the user. (SRE 124)

"Accession number <u>acc#</u> is recorded for <u>site</u> (<u>year</u>)."

5.17 If the response to the request for a command was:

Q, print a message at the terminal and repeat the request for a command.

"Use END instead of Q to end the SPECS session."

5.18 If the response to the request for a command was:

anything else, print a message at the terminal and repeat the request for a command.

"SPECS does not recognize your answer as a command."

6 If this was a normal session, close the files that are used only during normal sessions.

6.1 If the site-yr-acc file is open, close it.

6.2 If the transaction file is open, close it.

7 Execute shutdown procedures used by all SPECS sessions.

7.1 If SPECS does not require a batch update, ask the user whether he wants one.

7.1.1 Print a question at the terminal.

"Should a batch update be run?"

7.1.2 Wait for the user's response.

7.1.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter YES to request a batch update,

NO to request that one not be run, or

? for an explanation."

?, print an explanation at the terminal and repeat the question.

"SPECS is ending the current session. It wants to know whether to run a batch update before the next session. If you tell it to run a batch update, you must know the update password."

YES, remember that the update was requested.

NO, do nothing.

Q, print a message at the terminal and repeat the question.

"The SPECS session is ending, so there is nothing to quit."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your answer."

7.2 If the update was requested, ask the user for the update password, knowledge of which authorizes him to request batch updates.

7.2.1 Print a prompt at the terminal.

"Please enter the update password:"

7.2.2 Wait for the user's response.

7.2.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have your request for an update ignored, or

the correct password."

?, print an explanation at the terminal and repeat the prompt.

"You have asked SPECS to run a batch update. SPECS wants you to demonstrate your authority to make the request by telling it the update password."

Q, print messages at the terminal and on the session report, and remember not to run a batch update. (SRE 125)

"At your request, SPECS is ignoring your request for a batch update."

the correct password, print a message at the terminal and remember to run a batch update.

"A batch update will be run."

anything else (the first two times an incorrect password is given), print a message at the terminal and repet the prompt.

"Your response is not the correct password."

anything else (the third time), print messages at the terminal and on the session report, and remember not to run the batch update. (SRE 125)

"You have not given the correct password. SPECS is ignoring your request for a batch update."

7.3 Rewrite the progress record of the system parameters file, using the current value of the highest transaction number.

7.4 Write the current values of the records in the system parameters file on the session report.

7.4.1 Write a header for the summary of system parameters. (SRE 126)

7.4.2 Ask the user whether he wants the passwords printed on the report.

7.4.2.1 Print a question at the terminal.

"Do you want the passwords printed on the session report?"

7.4.2.2 Wait for the user's response.

7.4.2.3 If the response was:

:, print a menu at the terminal and repeat the question.

"Enter YES to request that the passwords be printed on the session report,

NO to request that the passwords not be printed, or

? for an explanation."

?, print an explanation at the terminal and repeat the question.

"SPECS is preparing to write the summary of the system parameters file on the session report. It wants to know whether you want the current passwords printed on the report. If you do, you must be able to supply the master password."

YES, remember that the master password is needed.

NO, remember that the master password is not needed.

Q, print a message at the terminal and repeat the question.

"The SPECS session is ending, so there is nothing to quit."

anything else, print a message at the terminal and repeat the question.

"SPECS cannot interpret your response."

7.4.3 If the master password is needed, and was not given earlier in the session, ask the user for it.

7.4.3.1 Print a message at the terminal.

"Please enter the master password:"

7.4.3.2 Wait for the user's response.

7.4.3.3 If the response was:

:, print a menu at the terminal and repeat the prompt.

"Enter ? for an explanation,

Q to have your request to print the passwords ignored, or

the correct password."

?, print an explanation at the terminal and repeat the prompt.

"You have asked SPECS to print the current paswords on the session report. SPECS wants you to demonstrate your authority to make the request by telling it the master password."

Q, print messages at the terminal and on the session report. (SRE 127)

"At your request, SPECS is ignoring your request to print the passwords."

the correct password, remember that the user knows it.

anything else (the first two times an incorrect password is given), print a message at the terminal and repeat the prompt.

"Your response is not the correct password."

anything else (the third time), print messages at the terminal and on the session report, and remember not to print the passwords. (SRE 127)

"You have not given the correct password. SPECS is ignoring your request to print the passwords on the session report."

7.4.4 If the master password was needed and the user gave it at some time during the session, print the passwords on the session report and a message at the terminal. (SRE 128)

"The passwords will be printed on the session report."

7.4.5 Print the validation constants on the session report. (SRE 129)

7.4.6 Print the progress information on the session report. (SRE 130)

7.5 Write a message at the terminal to tell the user that SPECS is ending.

"SPECS is ending. Please wait for a READY message before logging off at your terminal."

7.6 If earlier processing indicated that a batch update should be run, request it.

7.6.1 Submit a request for a batch update to the operating environment.

7.6.2 Write messages at the terminal and on the session report to tell the user that the request was made. (SRE 131)

"A batch update has been requested."

7.7 If the system parameters file is open, close it.

7.8 Write a concluding message on the session report. (SRE 132)

7.9 Close the session report file.

7.10 Return control to the operating environment, which is expected to print a READY message at the user's terminal.

Appendix F

BATCH-UPDATE PSEUDOCODE

F.1 INTRODUCTION

This appendix specifies an algorithm that could be used in a batch update, with particular attention to the master-file, transaction-file, and system-parameters-file records and update-report entries written by the update.

The batch-update pseudocode, like the interactive-session pseudocode (Appendix E), is expressed as a hierarchy of subdivided steps, each executed by performing all of its substeps. The steps are numbered according to the conventions described for the interactive-session pseudocode.

Steps that produce update-report entries have a "URE" followed by the number of an update-report entry listed in Appendix D.

If there are i insert-data records with a specimen number and m master-file records with the same specimen number, the algorithm given in this appendix uses $i + m + 2$ passes through the transactions for each specimen number in the transaction file. Since i, m, and the number of transactions for each specimen number are all likely to be very small, these passes probably contribute little to the expense of an update, but some could be eliminated by a clever implementation.

The implementor is encouraged to make any changes that he thinks improve the clarity, efficiency, or correctness of the batch-update algorithm, but he may not change the update-report entries or the records that are read or written.

F.2 PSEUDOCODE

```
1. Perform initialization.

    1.1 Open the update report file and write the update-report header on it. If the header can't be written, stop initialization.

        1.1.1 Open the update report file for writing.

        1.1.2 If the file was opened, write the update-report header. (URE 1)

    1.2 Prepare the system parameters file.

        1.2.1 Open the file for reading. If it can't be opened, write an update-report entry and stop initialization. (URE 2)

        1.2.2 Read the progress record. If it can't be read, write an update-report entry and stop initialization. (URE 3)

        1.2.3 Read the validation constants record. If it can't be read, write an update-report entry and stop initialization. (URE 4)

        1.2.4 Open the file for writing. If it can't be opened, write an update-report entry and stop initialization. (URE 5)
```

1.3 Open the site-yr-acc file. If it can't be opened, write an update-report entry and stop initialization. (URE 6)

1.4 Verify that the correct old master file (OMF) is provided.

  1.4.1 Open the file for reading. If it can't be opened, write an update-report entry and stop initialization. (URE 7)

  1.4.2 Read the first record, assuming that it is formatted as a master-file header. If it can't be read, write an update-report entry and stop initialization. (URE 8)

  1.4.3 Make sure that the first record is a master-file header by confirming that its "specimen type" is "M." If it isn't, write an update-report entry and stop initialization. (URE 9)

  1.4.4 If the value of the "master-file ID" field from the header record is not the same as the "file ID" from the progress record of the system parameters file, the master file is not the one to be updated. In this case, case, write an update-report entry and stop initialization. (URE 10)

1.5 Verify that the correct transaction file (XF) is provided.

1.5.1 Open the file for reading. If it cannot be read, write an update-report entry and stop initialization. (URE 11)

1.5.2 Read the first record on the file, assuming that it is formatted as a transaction-file header. If it cannot be read, write an update-report entry and stop initialization. (URE 12)

1.5.3 Make sure that the first record is a transaction-file header by confirming that its "specimen type" is "T". If it is not, write an update-report entry and stop initialization. (URE 13)

1.5.4 If the value of the "transaction-file ID" field from the header record is not the same as the "file ID" from the progress record of the system parameters file, the transaction file is not the one to be used in the update. In this case, write an update-report entry and stop initialization. (URE 14)

1.6 Prepare the new master file (NMF) by writing a master-file header on it.

  - 1.6.1 Open the file for writing. If it cannot be opened, write an update-report entry and stop initialization. (URE 15)

  - 1.6.2 Write a master-file header record for the NMF, using the "next transaction" value from the system parameters file as the master-file ID, the "master-file ID" value from the OMF as the old master-file ID, and the current date as the creation date. If the record can't be written, write an update report entry and stop initialization. (URE 16)

- 1.7 Prepare the new transaction file (NXF).

  - 1.7.1 Open the file for writing. If it cannot be opened, write an update-report entry and stop initialization. (URE 17)

  - 1.7.2 Write a transaction-file header record, using the "next transaction" value from the system parameters file as the transaction-file ID and the current date as the file creation date. If the record can't be written, write an update-report entry and stop initialization. (URE 18)

- 1.8 Increment the next transaction number to be assigned.

1.9 Read the next record on the OMF, if there is one. It is a specimen-data record.

1.10 Remember that there is no "previous specimen."

1.11 Sort the records in the XF into ascending order. The keys for the sort are:

1. Accession number
2. Sequence and continuation number
3. Specimen type
4. Transaction number.

If the records could not be sorted, write an update-report entry and stop initialization. (URE 19)

1.12 Find the first transaction to be processed by reading transaction-file records until either the end of the file or the first non-header record is found. Remember the ID's of the header records read during this step.

2 If the update-report file was successfully opened, write an update-report entry about the results of initialization. If initialization was stopped, end the batch update.

2.1 If the maser and transaction-file headers were read, write an update-report entry identifying the master and transaction files used in the update. (URE 20)

2.2 If initialization was stopped, write an update-report entry saying that no records were processed, close all opened files, and end the update. (URE 21)

3 Process transactions and OMF records until the end of the XF is found.

3.1 If the master-file key of the current OMF record comes before the master-file key of the current transaction in master-file sorting order, copy the OMF record to the NMF.

3.1.1 Remember the specimen represented by the current OMF record as the "previous specimen."

3.1.2 Write the current specimen-data record on the NMF.

3.2 If the mster-file key of the current OMF record does not come before the key of the current XF record, or if the end of the OMF has been found, process all OMF and XF records that have the same master-file key as the current XF record. The master-file key of the current XF record represents the "current specimen number."

3.2.1 Reconstruct the current specimen number from the master-file key, and print an update-report entry introducing it. (URE 22)

3.2.2 Read (and remember) all OMF and XF records for the current specimen number. (This involves reading until either the first record with a key greater than the current specimen number or the end of file is found for each file.)

3.2.3 Make a backward (in reverse sorting order) pass through the transactions for the current specimen number. This pass finds cancelled transactions and those with their validation overridden or restored, and also finds errors in transaction-modifying transactions. For each transaction, in reverse order:

3.2.3.1 If the current transaction has been marked as cancelled, or is an insert-data, delete-data, or change-data record, ignore it.

3.2.3.2 If the current transaction is a cancel-transaction record, process it.

3.2.3.2.1 If the transaction to be cancelled is ahead in the backward pass (that is, it was written in the XF before the current transaction was), and has not been marked as cancelled or as having its validation overridden or restored, mark it as cancelled by the current transaction.

3.2.3.2.2 If the transaction to be cancelled is ahead of the current transaction but has already been cancelled or had its validation overridden or restored, remember that the current transaction is superceded by the cancelling, overriding, or restoring transaction.

3.2.3.2.3 If the transaction to be cancelled is not ahead of the current transaction, consider the current transaction as erroneous because there is nothing to which it can be applied.

3.2.3.3 If the current transaction is a validation-restoration or validation-override record, process it.

3.2.3.3.1 If the transaction whose validation is to be affected is not an insert-data or change-data record, consider the current transaction as erroneous because it was applied to a transaction of an inappropriate type.

3.2.3.3.2 If the transaction whose validation is to be affected is ahead of the current transaction in the backward pass, and has not yet been marked as cancelled or as having its validation affected, mark it as having its validation restored or overridden by the current transaction.

3.2.3.3.3 If the transaction whose validation is to be affected is ahead of the current transaction, but has already been cancelled or had its validation overridden or restored, remember that the current transaction is superceded by the cancelling, restoring, or overriding transaction.

3.2.3.3.4 If the transaction whose validation is to be affected is not ahead of the current transaction, the current transaction is erroneous because there is nothing to which it can be applied.

3.2.4 Count the OMF records with the current specimen number.

3.2.5 Use a forward (in sorting order) pass through the transactions to list them, print messages about errors detected during the backward pass, and print messages about the effects of transaction-modifying transactions. This pass uses the count of records for the current specimen number to detect change-data and delete-data records for which the user did not give instructions about non-unique keys, but should have.

3.2.5.1 If the transaction is an insert-data record, list it. Increment the count of records with the current specimen number. (URE 23)

3.2.5.2 If the transaction is a change-data record, list it.

3.2.5.2.1 List the part of the record that precedes the new-field descriptions. (URE 24)

3.2.5.2.2 List each new-field description. (URE 25)

3.2.5.3 If the transaction is a delete-data record, list it. (URE 26)

3.2.5.4 If the transaction is a cancel-transaction record, list it. (URE 27)

3.2.5.5 If the transaction is a validation-restoration record, list it. (URE 28)

3.2.5.6 If the transaction is a validation-override record, list it. (URE 29)

3.2.5.7 If the transaction is an uncancelled change-data or delete-data record whose "transaction options" field has a value of -1 (the user gave no instructions for handling multiple specimen-data records with the same key), and the count of records with this specimen number is > 1, the current transaction is erroneous because it needed instructions for handling non-unique keys. In this case, write an update-report entry. (URE 30)

3.2.5.8 If the transaction is marked as superceded, print an update-report entry about it. (URE 31)

3.2.5.9 If the transaction is erroneous because there is nothing to which it can be applied, print an update-report entry about it. (URE 32)

3.2.5.10 If the transaction is erroneous because it was applied to a transaction of an inappropriate type, print an update-report entry about it. (URE 33)

3.2.5.11 If the transaction is cancelled, print an update-report entry about it. (URE 34)

3.2.5.12 If the transaction has its validation overridden, print an update-report entry about it. (URE 35)

3.2.5.13 If the transaction has its validation restored, print an update-report entry about it. (URE 36)

3.2.6 Make a pass through all of the specimen-data records with the current specimen number, applying all of the appropriate transactions to each specimen-data record. This pass is in two stages: the first for records from the OMF, and the second for the new specimen-data records created at the beginning of the stage. Step 3.2.6.1 gives directions for the first stage, step 3.2.6.2 gives directions for the second stage, and steps 3.2.6.3 through 3.2.6.13 are executed during both stages.

3.2.6.1 Perform steps 3.2.6.3 through 3.2.6.13 once for each specimen-data record from the OMF.

3.2.6.2 Use the insert-data transactions for this specimen number to create new specimen-data records, then apply the appropriate transactions to each of the new records.

3.2.6.2.1 Use each insert-data record to create a new specimen-data record, getting the values of the specimen-data fields from the insert-data fields with the same names.

3.2.6.2.2 Perform steps 3.6.2.3 through 3.6.2.13 for each new specimen-data record.

3.2.6.3 Consider the specimen described in the specimen-data record to be the "current specimen."

3.2.6.4 Write an update-report entry about the specimen-data record.

3.2.6.4.1 If the record is from the OMF, print an introduction to it. (URE 37)

3.2.6.4.2 If the record is from an insert-data record, print an introduction to it. (URE 38)

3.2.6.4.3 List the record. (URE 39)

3.2.6.5 Make a forward pass through all of the transactions, applying the appropriate ones to the specimen-data record.

3.2.6.5.1 If the specimen-data record was created during this batch update and the transaction number of the current transaction is lower than the number of the insert-data record from which the specimen-data record was created, ignore the current transaction.

3.2.6.5.2 If the current transaction is an insert-data, cancel-transaction, validation-override, or validation-restoration record, or if it is marked as cancelled or erroneous, ignore it.

3.2.6.5.3 Find out whether the current transaction is applicable to the current specimen-data record. A transaction is applicable if, and only if, one of these conditions is true:

1. No instructions were given for handling non-unique keys ("transaction options" is -1). (Recall that transactions without instructions for non-unique keys have already been detected as erroneous if there is more than one specimen-data record to which they could be applied.)
2. Instructions were given that the transaction should be applied to all specimen-data records for the specimen number ("transaction options" is 0).
3. The substring given in the "description" field of the transaction is included in the original value of the "description" field of the specimen-data record.

3.2.6.5.4 If the current transaction is an applicable delete-data record, consider the specimen-data record as deleted.

3.2.6.5.5 If the current transaction is an applicable change-data record and the specimen-data record is not deleted, make the changes indicated by each of the new-field descriptions in the current transaction.

3.2.6.5.5.1 If the field to be changed is not "location" or "description," replace the current value of the field by the new value from the change-data record.

3.2.6.5.5.2 If the "location" or "description" field is to be replaced, replace it by the new value from the change-data record. Replace the "location length" or "description length" by the value of the appropriate "field size" field of the change-data record.

3.2.6.5.5.3 If the "location" or "description" field is to be added to, add the new value. To add a value, concatenate a period followed by a blank to the front of the new value, then concatenate the new value to the end of the "location" or "description" field and recompute the location or description length. If the new length is > 127, truncate the new value to 127 characters, set the location or description length to 127, and write an update report entry.

3.2.6.5.5.3.1 If the "location" field was to be added to, print an update-report entry. (URE 40)

3.2.6.5.5.3.2 If the "description" field was to be added to, print an update-report entry. (URE 41)

3.2.6.5.5.4 Remember that the altered fields were most recently changed by the current transaction.

3.2.6.5.6 If the current transaction is an applicable change-data record and the specimen-data record is deleted, remember that the change-data record was applied. (It has no effect on a deleted record.)

3.2.6.6 Print an update-report entry listing the transactions that were applied. (URE 42)

3.2.6.7 If the specimen-data record was not deleted, perform multiple-field validation.

3.2.6.7.1 If the value of the "description" or "specimen type" field was changed, perform type-description validation.

3.2.6.7.1.1 Apply the algorithm in Appendix C.

3.2.6.7.1.2 If validation fails, print an update-report entry saying that the description and specimen type are inconsistent. Consider the "description" and "specimen type" fields that were changed for the current specimen during this batch update as erroneous. (URE 43)

3.2.6.7.2 If the value of the "description," "specimen type," "count," "count type," "weight," or "weight type" field was changed, perform kind-count-weight validation.

3.2.6.7.2.1 Apply the algorithm in Appendix C.

3.2.6.7.2.2 If validation fails, tell the user why.

3.2.6.7.2.2.1 If the soil sample or washings are too heavy, write an update-report entry. (URE 44)

3.2.6.7.2.2.2 If a count is too high, write an update-report entry. (URE 45)

3.2.6.7.2.2.3 If a density is outside of the expected range, write an update-report entry. (URE 46)

3.2.6.7.2.3 If validation fails, consider all of the "description," "specimen type," "count," "count type," "weight," and "weight type" fields whose values for the current specimen were changed during this batch update as erroneous.

3.2.6.7.3 If the value of the "site," "year," or "accession number" field was changed, perform site-yr-acc validation.

3.2.6.7.3.1 Apply the algorithm in Appendix C.

3.2.6.7.3.2 If validation fails, tell the user why.

3.2.6.7.3.2.1 If the site and year were not found in the site-yr-acc file, print an update-report entry. (URE 47)

3.2.6.7.3.2.2 If the accession number is inconsistent with the previous entries in the file, print an update-report entry. (URE 48)

3.2.6.7.3.3 If the validation fails, consider all of the "site," "year," and "accession number" fields whose values for the current specimen were changed during this batch update as erroneous.

3.2.6.8 If the current specimen was not deleted, perform sequence checking for depth, comparing the "current" and "previous" specimens.

3.2.6.8.1 Apply the algorithm in Appendix C if both of these conditions are true:

1. There is a "previous specimen."
2. Either one or more of the "site," "year," "grid-square type," "N-S coord," "L-R half," "E-W coord," "unit," "unit type," "depth," "depth type," "accession number," "sequence and continuation number," or "specimen type" fields of the current specimen-data record was changed,

or the previous specimen was read from the OMF and the current specimen was created from an insert-data transaction during this batch update.

3.2.6.8.2 If validation fails, print an update-report entry to say that depth types are unsorted or mixed. (URE 49)

3.2.6.8.3 If validation fails, consider all of the "site," "year," "grid-square type," "unit," "unit type," "depth," "N-S coord," "L-R half," "E-W coord," "accession number," "sequence and continuation number," "specimen type," and "depth type" fields whose values were changed for the current specimen as erroneous.

3.2.6.9 If erroneous fields were found, restore their values from the beginning of the batch update, unless validation was overridden for the transactions that set the erroneous values or the "erroneous" values had already reverted to their original values from an OMF record (records from the OMF are assumed to be correct.)

3.2.6.9.1 For each erroneous value:

3.2.6.9.1.1 If the transaction that last set the value had its validation overridden, consider the value correct.

3.2.6.9.1.2 If the value is not considered correct and was given by an insert-data transaction, reject the current specimen-data record.

3.2.6.9.1.3 If the value is not considered correct and the transaction that last set the value was a change-data record, replace the erroneous value by its "original" value (the value of the same field from the OMF or the insert-data record from which the specimen-data record was created during this update).

3.2.6.9.2 Write an update-report entry about each erroneous value in records that were not rejected.

3.2.6.9.2.1 If the validation was overridden, tell the user the value was accepted. (URE 50)

3.2.6.9.2.2 If the new value was replaced by an original one, tell the user about the value. (URE 51)

3.2.6.10 If erroneous values were restored and the specimen-data record was neither rejected nor deleted, repeat validation for the current values by restarting step 3.2.6.7. During the repetition, consider only the values that were replaced during the most recent execution of step 3.2.6.9 as changed.

3.2.6.11 Find out whether a change-data transaction was successfully applied to a specimen-data record that had information from a "unit," "grid square," or "depth" input field stored in the "location" field. (If it was, the information in the "location" field may no longer be accurate.) (A successful change is one that either passed all validation algorithms applied to it or was considered correct because validation was overridden for the transaction that made the change.)

3.2.6.11.1 If "unit-type" in the specimen-data record was "O" at the beginning of the batch update, information from the "unit" input field was successfully changed, and the "location" field was not successfully replaced, write an update-report entry warning the user that the location information may be inaccurate. (URE 52)

3.2.6.11.2 If "depth-type" in the specimen-data record was "O" at the beginning of the batch update, information from the "depth" input field was successfully changed, and the "location" field was not successfully replaced, write an update-report entry warning the user that the location information may be inaccurate. (URE 53)

3.2.6.11.3 If "grid-square type" in the specimen-data record was "O," "A," or "P" at the beginning of the batch update, information from the "grid" input field was successfully changed, and the "location" field was not successfully replaced, write an update-report entry warning the user that the location information may be inaccurate. (URE 54)

3.2.6.12 Write new master-file and transaction-file records, and complete processing of changed specimen numbers.

3.2.6.12.1 If both of these conditions are true, write the current specimen-data record on the NMF:

1. The record was not deleted or rejected.

2. None of the "accession number," "sequence and continuation number," and "specimen type" fields of the record was successfully changed during the batch update.

3.2.6.12.2 If the specimen-data record was written, and is the last with the master-file key to be processed, remember it as the new "previous specimen".

3.2.6.12.3 If the "accession number," "sequence and continuation number," or "specimen type" field was successfully changed, the specimen-data record for the current specimen must be inserted at some other place in the master file. Write a new insert-data transaction to add the record in the correct place during the next batch update.

3.2.6.12.3.1 Assign the values of the specimen-data fields to the fields with the same names in a new insert-data record.

3.2.6.12.3.2 Set the "transaction number" field of the insert-data record to the next transaction number to be assigned.

3.2.6.12.3.3 Increment the next transaction number to be assigned.

3.2.6.12.3.4 Write the new insert-data record on the NXF.

3.2.6.13 Write an update-report entry about the results of processing.

3.2.6.13.1 If the specimen-data record was not deleted, print the values of the fields in the record.

3.2.6.13.1.1 Write an introduction on the update report. (URE 55)

3.2.6.13.1.2 Write an update-report entry about the values of the fields. (URE 39)

3.2.6.13.2 Write an update-report entry about the disposition of the specimen-data record.

3.2.6.13.2.1 If the record was written on the NMF, say so. (URE 56)

3.2.6.13.2.2 If the record was used to generate a new insert-data record because of a change in the specimen number, say so. (URE 57)

3.2.6.13.2.3 If the record was deleted, say so. (URE 58)

3.2.6.13.2.4 If the record was rejected, say so. (URE 59)

3.2.7 Print update-report entries about any change-data and delete-data transactions that were not appled to any specimen-data record. (URE 60)

3.2.8 Print a concluding message for the specimen number on the update report. (URE 61)

3.3 If the end of the XF has not been found, restart step 3.1 to process the records for the next specimen.

4 If the end of the OMF has not been found, copy the rest of the OMF records directly to the NMF.

5 Perform shutdown.

5.1 Print an update-report entry saying that processing is complete. Give counts of records added, deleted, changed, rejected, and used to generate new input-data records, and the total number of NMF records. (URE 62)

5.2 Update and rewrite the progress record of the system parameters file.

5.2.1 Set "file ID" to the "master file ID" of the NMF (the value of "next transaction" at the beginning of the update). This establishes the NMF from this batch update as the OMF for the next one. Write an update-report entry to tell the user that the current master file version has changed. (URE 63)

5.2.2 Rewrite the progress record, using the current values of "file ID" and "next transaction." If it cannot be written, write an update-report entry. (URE 64)

5.3 Write the current progress information and validation constants on the update report.

5.3.1 Write a header for the system parameters report. (URE 65)

5.3.2 Write the current validation constants. (URE 66)

5.3.3 Write the current progress information. (URE 67)

5.4 Close the system parameters file, OMF, NMF, XF, NXF, and site-yr-acc file.

5.5 Close the update report file.

5.6 Return control to the operating environment.

Appendix G

SPECIMEN-CATALOG DATA

This appendix lists the specimens whose specimen-catalog entries were studied for this paper.

Acc #	Site	Year	Specimens	Comments
1901	MkV74	1961	1 - 3	
1902	MkV75		1 - 3	
1903	MkV76		1 - 2	
1904	MkV77		1 - 4	
1905	MkV78		1 - 3	
1906	MkV79		1 - 2	
1907	MkV80		1	
1908	MkV81		1	
1909	MkV82		1 - 2	
1910	MkV83		1 - 2	
1911	MkV84		1	
1912	PtV6	1964	1 - 7	
1913	PtV7		1 - 9	
1914	PtV8		1 - 4	
1915	PtV2		1 - 2	
1916	BfV51		1 - 7	
1930	NC-G	1956	1 - 51	
1931	India	1964	1 - 7	Ethnographic

1932	ChV4	1964	1 - 11	New Hope Survey
	ChV5		12	"
	ChV8		13 - 22	"
	ChV15		23 - 25	"
	ChV16		26 - 34	"
	ChV18		35 - 45	"
	ChV20		46 - 48	"
	ChV25		49 - 59	"
	ChV26		60 - 62	"
	ChV27		63 - 73	"
	ChV28		74 - 79	"
	ChV29		80 - 81	"
	ChV30		82 - 90	"
	ChV31		91 - 92	"
	ChV32		93 - 102	"
	ChV33		103 - 109	"
	"		675 - 684	"
	ChV37		110 - 114	"
	ChV38		115 - 122	"
	ChV39		123 - 125	"
	ChV40		126 - 128	"
	ChV41		129 - 133	"
	ChV42		134 - 140	"
	ChV43		141 - 142	"
	ChV44		143 - 153	"
	ChV45		154 - 161	"
	ChV46		162 - 168	"

1932	ChV48	1964	169 - 172	New Hope Survey
	ChV49		173 - 175	"
	ChV50		176 - 186	"
	ChV51		187 - 194	"
	ChV52		195 - 196	"
	ChV53		197 - 205	"
	ChV54		206 - 211	"
	ChV55		212 - 215	"
	ChV56		216 - 220	"
	ChV57		221 - 230	"
	ChV58		231 - 239	"
	ChV59		240 - 244	"
	ChV60		245 - 252	"
	ChV61		253 - 257	"
	ChV62		258 - 259	"
	ChV63		260 - 262	"
	ChV64		263 - 267	"
	ChV65		268 - 272	"
	ChV66		273 - 277	"
	ChV67		278 - 281	"
	ChV68		282 - 285	"
	ChV69		286 - 287	"
	ChV70		288 - 295	"
	ChV71		296 - 299	"
	ChV72		300 - 307	"
	ChV73		308 - 313	"
	ChV74		314 - 318	"

1932	ChV75	1964	319 - 324	New Hope Survey
	ChV76		325 - 329	"
	ChV77		330 - 333	"
	ChV78		334 - 339	"
	ChV79		340 - 345	"
	ChV80		346 - 347	"
	ChV81		348 - 349	"
	ChV82		350 - 353	"
	ChV83		354 - 357	"
	ChV84		358 - 363	"
	ChV85		364 - 389	"
	ChV86		390 - 391	"
	ChV87		392 - 395	"
	ChV88		396 - 403	"
	ChV89		404 - 406	"
	ChV90		407 - 409	"
	ChV91		410 - 413	"
	ChV92		414 - 416	"
	ChV93		417 - 420	"
	ChV94		421 - 424	"
	ChV95		425 - 432	"
	ChV96		433 - 439	"
	ChV97		440 - 441	"
	ChV98		442 - 449	"
	ChV99		450 - 454	"
	ChV100		455 - 462	"
	ChV101		463 - 468	"

1932	ChV102	1964	469 - 473	New Hope Survey
	ChV103		474 - 479	"
	ChV104		480 - 484	"
	ChV105		485 - 490	"
	ChV106		491 - 493	"
	ChV107		492 - 493	"
	ChV108		494 - 495	"
	ChV109		496	"
	ChV110		497 - 499	"
	ChV111		500 - 506	"
	ChV112		507 - 510	"
	ChV113		511 - 512	"
	ChV114		513 - 514	"
	ChV115		515	"
	ChV116		516 - 519	"
	ChV117		520 - 522	"
	ChV118		523 - 524	"
	ChV119		525 - 528	"
	ChV120		529 - 530	"
	ChV121		531 - 536	"
	ChV122		537 - 539	"
	ChV123		540 - 545	"
	ChV124		546 - 551	"
	ChV125		552	"
	ChV126		553 - 556	"
	ChV127		557	"
	ChV128		558 - 559	"

1932	ChV129	1964	560 - 561	New Hope Survey
	ChV131		562 - 568	"
	ChV132		569 - 570	"
	DhV40		571 - 574	"
	DhV30		575 - 580	"
	DhV31		581 - 588	"
	DhV32		589 - 593	"
	DhV33		594 - 602	"
	DhV34		603 - 606	"
	DhV35		607 - 615	"
	ChV134		616 - 618	"
	ChV135		619 - 622	"
	ChV136		623 - 624	"
	ChV137		625 - 626	"
	ChV139		627 - 630	"
	ChV141		631 - 637	"
	ChV142		638 - 644	"
	ChV143		645 - 651	"
	ChV144		652 - 654	"
	DhV20		655 - 658	"
	DhV36		659 - 661	"
	DhV37		662 - 664	"
	DhV38		665 - 670	"
	DhV39		671 - 672	"
	ChV145		685 - 691	"
1933	TrV1	1964	1 - 5	Transylvania County, Sites 1 - 150

1933	TrV2	1964	6 - 7	Transylvania County, Sites 1 - 150
	TrV3		8 - 10	"
	TrV4		11 - 14	"
	TrV7		15 - 17	"
	TrV8		18 - 20	"
	TrV9		21 - 23	"
	TrV10		24 - 27	"
	TrV11		28 - 29	"
	TrV12		30 - 31	"
	TrV13		32	"
	TrV14		33 - 34	"
	TrV15		35 - 36	"
	TrV16		37 - 42	"
	TrV17		43 - 43/1	"
	TrV18		44 - 45	"
	TrV19		46 - 47	"
	TrV20		48 - 50	"
	TrV21		51 - 54	"
	TrV22		55 - 58	"
	TrV23		59 - 60	"
	TrV24		61 - 62	"
	TrV25		63 - 64	"
	TrV26		65 - 67	"
	TrV27		68 - 70	"
	TrV28		71 - 73	"
	TrV29		74 - 77	"

1933	TrV30	1964	78 - 80	Transylvania County, Sites 1 - 150
	TrV31		81 - 83	"
	TrV32		84 - 85	"
	TrV33		86 - 92	"
	TrV34		93 - 96	"
	TrV35		97 - 98	"
	TrV36		99 - 100	"
	TrV37		101	"
	TrV38		102 - 104	"
	TrV39		105 - 111	"
	TrV40		112 - 114	"
	TrV41		115 - 117	"
	TrV42		118 - 123	"
	TrV43		124 - 125	"
	TrV44		126 - 128	"
	TrV45		129 - 130	"
	TrV46		131 - 132	"
	TrV47		133 - 136	"
	TrV48		137 - 140	"
	TrV49		141 - 143	"
	TrV50		144	"
	TrV51		145 - 147	"
	TrV52		148 - 151	"
	TrV53		152 - 153	"
	TrV54		154 - 155	"
	TrV55		156 - 157	"

1933	TrV56	1964	158 - 159	Transylvania County, Sites 1 - 150
	TrV57		160 - 162	"
	TrV58		163 - 164	"
	TrV59		165 - 167	"
	TrV60		168 - 169	"
	TrV61		170 - 172	"
	TrV62		173	"
	TrV63		174 - 177	"
	TrV64		178 - 181	"
	TrV65		182 - 184	"
	TrV66		185	"
1938	CeV15	1964	1 - 2	
	"		167 - 532	
1939	GhV2	1964	1 - 4	Graham County, 1964 Survey
	GhV4		5 - 6	"
	GhV5		7 - 8	"
	GhV6		9	"
	GhV7		10 - 11	"
	GhV8		12 - 14	"
	GhV9		15 - 16	"
	GhV10		17 - 18	"
	GhV11		19 - 23	"
	GhV12		24 - 25	"
	GhV13		26	"
	GhV14		27 - 28	"

1939	GhV15	1964	29 - 30	Graham County, 1964 Survey
	GhV16		31 - 32	"
	GhV17		33 - 36	"
	GhV18		37 - 38	"
	GhV19		39 - 41	"
	GhV20		42 - 43	"
	GhV21		44 - 45	"
	GhV22		46 - 49	"
	GhV23		50 - 54	"
	GhV24		55 - 57	"
	GhV25		58 - 59	"
	GhV26		60 - 61	"
	GhV27		62	"
	GhV28		63 - 64	"
	GhV29		65 - 66	"
	GhV30		67 - 69	"
	GhV31		70 - 72	"
	GhV32		73 - 74	"
	GhV33		75	"
	GhV34		76 - 78	"
	GhV35		79	"
	GhV36		80 - 82	"
	GhV37		83 - 84	"
	GhV38		85 - 86	"
	GhV39		87 - 88	"
1940	MlV3	1964	1 - 128	

1941	Cy01	1964	1 - 6	Clay County, 1964 Survey
	CyV3		7 - 19	"
	CyV5		20 - 21	"
	CyV8		22 - 24	"
	CyV9		25 - 29	"
	CyV10		30 - 31	"
	CyV11		32 - 33	"
	CyV12		34 - 36	"
	CyV13		37 - 38	"
	CyV14		39 - 41	"
	CyV15		42 - 44	"
2241	BnV29	1972	1 - 434	Warren Wilson Site
	"		1869 - 2197	"
	"		2460 - 2601	"
2317	SkV1a	1976	1 - 1372	

REFERENCES

1. IBM, Introduction to IBM Direct-Access Storage Devices and Organization Methods. IBM, 1978, Chapter 9.

2. IBM, OS PL/I Checkout and Optimizing Compilers: Language Reference Manual (GC33-0009-4). IBM, 1976.

3. Helwig, J.T., SAS Introductory Guide. SAS Institute Inc., Cary, NC, 1978.

4. Nie, N.H., Hull, C.H., Jenkins, J.G., Steinbrenner, K., Brent, D.H., SPSS: Statistical Package for the Social Sciences. McGraw-Hill Book Company, New York, 1975.

5. SAS Institute Inc., SAS Programmer's Guide, 1979 edition. SAS Institute Inc., Raleigh, NC, 1979.

6. SAS Institute Inc., SAS Supplemental Library User's Guide, 1980 edition. SAS Institute Inc., Cary, NC, 1980.

7. SAS Institute Inc., SAS User's Guide, 1979 edition. SAS Institute Inc., Cary, NC, 1979.

8. The University of North Carolina at Chapel Hill, Record of the University of North Carolina at Chapel Hill: The Graduate School (USPS 651-960, number 900). The University of North Carolina at Chapel Hill, Chapel Hill, NC, April, 1980, p. 87.

"Enter ? for an explanation,

Q to request that the SHOW-ACC command be ignored, or

a year."

?, print an explanation at the terminal and repeat the prompt.

"You have asked SPECS to show the accession number for a specified site and year. SPECS could not find the year in your command, and is asking you to repeat the information."

Q, print a message at the terminal and stop executing the SHOW-ACC command.

"At your request, SPECS is ignoring the SHOW-ACC command."

anything else, assume it is the year.

5.16.4.4 Use the algorithm in Appendix C to validate the assumed year. If the year information is missing or malformed, print a message at the terminal and get another year by restarting step 5.16.4.

"SPECS cannot interpret your response as a year."