

University of Mississippi

eGrove

---

Guides, Handbooks and Manuals

American Institute of Certified Public  
Accountants (AICPA) Historical Collection

---

1983

## Auditing & EDP

Gordon B. Davis

Donald L. Adams

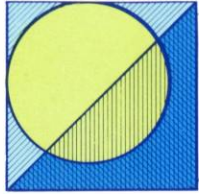
Carol A. Schaller

Follow this and additional works at: [https://egrove.olemiss.edu/aicpa\\_guides](https://egrove.olemiss.edu/aicpa_guides)



Part of the [Accounting Commons](#)

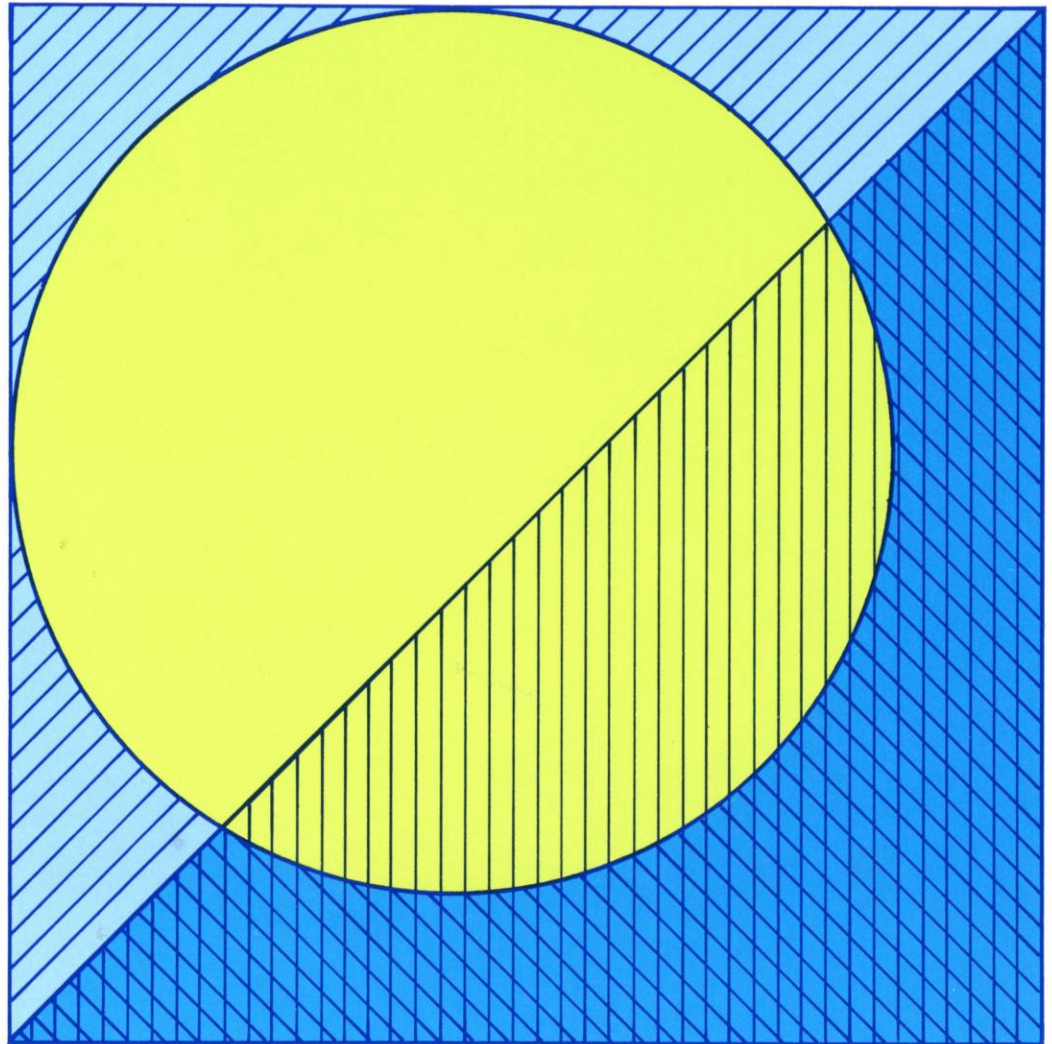
---



# AUDITING & EDP

Second Edition

GORDON B. DAVIS, CPA • DONALD L. ADAMS, CPA • CAROL A. SCHALLER, CPA



**AUDITING & EDP**  
Second Edition  
DAVIS • ADAMS • SCHALLER

AICPA

# **AUDITING & EDP**

**Second Edition**



***GORDON B. DAVIS, CPA***

**HONEYWELL PROFESSOR OF MANAGEMENT INFORMATION SYSTEMS  
UNIVERSITY OF MINNESOTA**

***DONALD L. ADAMS, CPA***

**AMERICAN INSTITUTE OF CERTIFIED PUBLIC ACCOUNTANTS**

***CAROL A. SCHALLER, CPA***

**ERNST & WHINNEY**

**AMERICAN INSTITUTE OF CERTIFIED PUBLIC ACCOUNTANTS**

Copyright © 1983 by the  
American Institute of Certified Public Accountants, Inc  
1211 Avenue of the Americas, New York, N Y 10036-8775  
1 2 3 4 5 6 7 8 9 0 F&A 8 9 8 7 6 5 4 3

# Contents

## Preface

## I. The Audit and Computer Data Processing

### 1. The Auditor and the Computer 3

The Computer in Data Processing	3
Trends in Application of Computers by Organizations	3
The Computer and Client Services by CPAs	4
Audit Practice and the Computer	4
Tax Practice and the Computer	4
Accounting Services and the Computer	5
Management Advisory Services	5
Accounting Practice Research	5
The Effect of the Computer on Data Processing Controls	5
Organizational Control Procedures	6
Processing Controls Necessary	6
Computer System Alternatives to Manual Controls	6
The Study and Evaluation of Internal Accounting Control When a Computer is Used	7
Special Reviews of Computer Data Processing	8
Data Processing Management Review	8
Application Postimplementation Review	8
Computer Processing Continuation Provisions Review	8
Privacy Compliance Review	9
Relationship of Independent Auditors to Internal Auditors	9
Auditors' Need for Competence in Computer Data Processing	9
Summary	10

## Contents

### **2. Planning and Applying Audit Procedures for Computer Data Processing 11**

Classification of Controls in Computer Data Processing	11
General Scope for Audit Procedures in an EDP Environment	13
Accounting Control Within the Required Scope	13
Computer Operations Continuation Provisions An Optional Extension of Scope	15
General Controls and Application Controls	16
General Controls	16
Application Controls	18
Risk and Exposure Analysis in Audit Planning	19
EDP Function Risk and Exposure Analysis	19
Application Level Risk and Exposure Analysis	20
Transaction Level Risk and Exposure Analysis	20
Data Item Analysis	21
Overview of the Audit Program as it Relates to Computer Data Processing	21
Unique Characteristics of Audits in an EDP Environment	21
Steps in the Audit Program Related to Computer Data Processing	21
The Study and Evaluation of Internal Accounting Control in EDP Systems	22
Substantive Testing	22
Scheduling Computer-Related Audit Activities	22
Summary	24

### **3. The Study and Evaluation of Internal Accounting Control in Computer Data Processing 27**

Steps in the Process	27
Preliminary Phase of the Review	29
<i>General Understanding of the Data Processing System for Accounting Records, 29</i>	
<i>General Understanding of the Extent to Which EDP Is Used in Significant Accounting Applications, 29</i>	
<i>Outline of the Basic Structure of Accounting Control for EDP, 30</i>	
<i>Preliminary Assessment, 30</i>	
Completion of the Review	31
Tests of Controls	31
Evaluation of Accounting Control in the Data Processing System	32
Applying the Distinction Between the Two Phases of the Review	33
<i>Questions at the Preliminary Phase, 34</i>	
<i>Questions at Completion of the Review, 34</i>	
The Optional Review of Provisions for EDP System Continuation	34
Summary	35

## II. General Controls for Computer Data Processing

### 4. Organization and Management of the EDP Function 39

Plan of Organization	39
Job Descriptions	39
Segregation of Functions	40
Segregation of Functions Between EDP and Users	41
Typical Organizational Charts	41
The EDP Control Function	45
Outside Data Processing Control	45
Processing Control Functions	45
Librarian Control Function	46
Data Administration Function	46
Management of Computer Installations	47
Personnel	47
Operating Procedures	48
Standards	48
Master Plan	50
Management and Control of Application Systems Development	50
Application Development Life Cycle	51
Quality Assurance in Programming	51
User and Auditor Involvement in the Application Development Life Cycle	53
Alternatives to Traditional Development Life Cycle	53
Testing of Application Systems During Development and Implementation	54
Program/Module Tests	54
String Tests	54
System Tests	54
Pilot Tests	55
Parallel Tests	55
Auditor's Role in Testing	55
Relationship of Organization and Management Practices to Accounting Controls	55
Summary	57

### 5. Documentation of the Application 59

Types of Documentation	60
System Documentation	60
Job Documentation	60
<i>Job Narrative, 60 Run-to-Run Flowchart, 61 Program Summary, 61</i>	
<i>Source Document Summary, 61 File Summary, 61 Output Summary, 61</i>	

## Contents

Program Documentation	62
<i>Program Narrative, 62 Program Structure and Logic Description, 62</i>	
<i>Source Code Listing, 62 Test Results or Sample Output, 62</i>	
<i>Change Notice, 62</i>	
File Documentation	64
Operations Documentation	67
User Documentation	67
Software Documentation Aids	69
Flowchart Packages	69
Cross-Reference Listings	72
File Description Generators	72
Formatting	72
Librarian Packages	75
Minimum Documentation and the Auditor	75
Systems Documentation	77
Program Documentation	78
Operations Documentation	78
The Importance of Documentation for Audits	79
Summary	79

## **6. Hardware Features for Control Over Equipment Malfunctions 83**

How the Equipment Can Malfunction	83
Failure in Electronic Components	83
Failure in Mechanical Operation	84
Equipment Controls	84
Redundant Character Check	84
Duplicate Process Check	84
Echo Check	84
Validity Check	84
Equipment Check	85
Central Processor and Storage Hardware Controls	85
Parity Check	85
Automatic Retry and Diagnosis	86
Validity Check	86
Storage Protection	87
Card Reader Hardware Controls	87
Printer Controls	87
Magnetic Tape Hardware Controls	88
Parity Check for Magnetic Tape	90
Error Correction Codes	90
Tape Unit Monitoring	91
Disk Storage Hardware Controls	91



Parity Check 91  
 Error Correction Codes 93  
 Retry Facility 93  
 Address Checking 93  
 Error Controls for Data Communications 93  
   Causes of Data Communications Errors 94  
   Methods of Error Detection and Correction 94  
 Other Input/Output Devices 95  
   Magnetic Ink Character Reader 95  
   Optical Character Recognition Equipment 95  
 Power Supply 96  
 Hardware Controls and the Audit 96  
 Summary 97

**7. System Software and Database Management Systems 99**

Operating Systems 99  
   Functions and Control Features 99  
     *Managing Concurrent Processing, 100 Scheduling Processing, 100*  
     *Allocating Resources, 100 Monitoring and Controlling Processing, 101*  
     *Handling Errors, 101 Maintaining Logs and Computer Accounting Infor-*  
     *mation, 102 Aiding Data Storage, Maintenance, and Access, 102*  
     *Aiding Program Storage, Maintenance, and Access, 102*  
   Audit Considerations for Operating Systems 102  
 Utility Programs 103  
   Examples of Utilities Used for Control 103  
   Control Concerns for Utility Programs 104  
   Audit Considerations for Utility Programs 104  
 Database Management Systems 104  
   The Database Concept 105  
   The Components of a Database Management System 105  
   How the DBMS Works 106  
   Control Advantages of a Database Management System 107  
     *Data Consistency, 107 Timeliness and Availability of Data, 107 In-*  
     *creased Data Integrity, 108 Increased Data Security and Control Over*  
     *Access to Data, 108 Improved Documentation, 108*  
   Control Concerns in a Database Management System 108  
     *Concentration of Data, 108 Single Recording of a Data Item, 108*  
     *Complexity of the Storage Structure and Software, 109 Centralization*  
     *of Duties, 109*  
   The Database Administrator 109  
     *Segregation of Duties Within the DBA Function, 110 Segregation From*  
     *User Functions, 110 Segregation From Computer Operation Func-*  
     *tions, 110 Segregation From Systems Development and Programming*

## Contents

*Functions, 110 Maintenance of Logs and Reports of Database Activities, 110 Monitoring of Data Administration Activities, 110 Care in the Selection of DBA Personnel, 111*

Audit Considerations for Database Management Systems 111

Summary 111

### **8. Safeguarding the Availability, Access, and Use of Computer Facilities, Programs, and Data 113**

Safeguarding Computer Facilities 114

Location and Construction of the Computer Room 114

Fire Prevention and Detection and Fire-Fighting Techniques 114

Insurance 115

Housekeeping and Preventive Maintenance 115

Physical Safeguards for Files 116

Control of Access to the Computer Room 119

Safeguarding Computer Programs and Data 120

Librarian Function 120

Passwords and Other Identification Techniques 121

Encryption 123

Program Change Process 123

Operator Procedures to Control the Use of Programs and Data 124

Library Software and Other Security Software 125

Backup, Recovery, and Retention Procedures 126

Physical Backup Facilities 126

Backup of Programs 127

Retention and Recovery Plans for Data 127

*Source Documents, 127 Magnetic Tape Files, 128 Disk Files, 129 Dumps, 129*

Summary 131

## **III. Processing Methods and Controls for Applications**

### **9. Control Implications of Data Processing Methods 135**

The Basic Computer Data Processing Cycle 135

Data Capture and Data Entry 135

Data Preparation 138

Input 139

Input Validation 139

Processing 139

Output 140

Classification of Computer Files 140

File Storage Devices	140
File Organization	141
Database and Database Management Systems	141
Multiple-Computer Access to Files	141
Processing Methods	142
Terminal Entry/On-Line Processing	142
Terminal Entry/Batch Processing	146
Batch Entry/Batch Processing	148
Summary	149

**10. Application Controls: Authorization, Data Validation, Error Handling, and File Quality Maintenance 153**

Authorization	153
Methods for Validating Data	155
Logic Tests	155
Check Digits	156
Echo Checks	157
Stored Data Comparison	157
Control Totals	157
Output Review	158
Generalized Input Data Validation	158
Control Over Error Notification, Correction, and Resubmission	161
Timing of Corrections	161
Validity of Corrections	161
Control Over Resubmission	161
Circulating Error File	162
File Quality Maintenance Procedures	164
File Quality Reports	164
File Detail Reviews	164
Summary	164

**11. Application Controls: Data Preparation, Input, Processing, and Output 167**

Potential Sources of Errors in Data Preparation, Input, or Entry	167
Control Over Batch Data Preparation	168
Procedural Controls, Editing, and Data Review	168
Preparing Batches of Transaction Documents	168
Transmittal Controls and Route Slips	169
Key Entry to Machine-Readable Form	169
Key Verification	170
Machine-Readable Data Input Preparation	170

## Contents

Control of Input or Direct Entry	171
Batch Input	171
Terminal Entry and Batch Processing	171
Immediate Terminal Entry and Immediate Processing of Individual Transactions	172
Sources of Processing Errors	173
Use of an Incorrect Version of the Program	173
Use of a Wrong File or Record in Processing	173
Incorrect Value in Internal Tables	174
Wrong Default Value	174
Input of Incorrect Program Parameters	174
Precision and Rounding Errors	174
Incorrect or Incomplete Processing Logic	175
Programmed Processing Error Controls	176
Limit and Reasonableness Test (Range Test)	176
Sequence Test	176
Explicit Class Identification	176
Crossfooting Test	176
Control Totals	177
Process Tracing Data	177
Error Correction Screening	177
Processing Errors and Operating Controls	178
Breakpoints in Processing	178
Backup Provisions	178
File Usage Controls	178
Database Access Controls	179
Run-to-Run Controls	179
Control Over Output	179
Distribution Control	179
Output Reviews	180
Summary	180

## **12. The Audit Trail in Computer Data Processing 183**

Definition of an Audit Trail	183
Elements of an Audit Trail	186
Transaction Source Record	186
Reference and Control List	187
Process Tracing Data	189
Transaction Source References	190
Examples of Audit Trail Methods	193
Hourly Payroll Processing Using Batch Processing	193

On-Line Sales Order Entry	193
Aging of Accounts Receivable	195
A Review Checklist for Evaluating the Audit Trail in an Application	197
Summary	197

## IV. Computer Audit Tools and Techniques

### 13. Computer Audit Tools and Techniques Used After Processing: Audit Software Systems 201

Functions Performed by Audit Software	201
Data Access	202
Selection and Statistical Functions	202
Computation	205
Comparison	205
Record Handling	205
Output	205
Miscellaneous Functions	207
Types of Generalized Audit Software	207
Coding Versus Checklist	207
Precompilers Versus Interpreters	207
Selecting a Generalized Audit Software Package	211
Definition of Needs	211
Identification of Computer Resources	211
Level of Auditor Expertise Required	213
<i>General EDP Knowledge, 213 Training in the Use of the Audit Software Package, 213</i>	
Functions Required	214
Vendor Support	214
Case Example of the Use of Audit Software	214
Summary	217

### 14. Computer Audit Tools and Techniques Used After Processing: Testing Files, Static Testing of Programs, and Simulation Testing 219

Testing of Files	220
Query Facilities of File Management and Database Management Systems	220
Customized Audit Software	220
Modification of Existing Client Programs	221
Utility Software	221

## Contents

Audit Software Applied to Small Computer System File Testing	222
Specialized Industry Software	222
Static Testing of Programs	224
Test Data	224
Test Data Generators	227
Optimizer Packages to Evaluate Test Data	227
Simulation or Parallel Processing to Test Application Processing	231
Summary	233

### **15. Computer Audit Tools and Techniques Used After Processing: Review of System Operations, Code Review, and Other Software Tools 235**

Review of System Operations	235
Logging Facilities	235
Job Accounting Facilities	236
Review of Program Code	240
An Outline of the Code Review Process	240
Software Review Aids	244
<i>Flowcharting Software, 244 Cross-Reference Systems, 244</i>	
<i>Record Layout Software, 245 Optimizers, 245</i>	
Audit Control of Program Changes	245
<i>Change Detection, 245 Librarian Software, 246</i>	
Third-Party Code Review	246
Other Software Tools	247
Vendor Libraries	247
Software Catalogs	248
Time-Sharing Services	248
Summary	249

### **16. Computer Audit Techniques Used Concurrently With Processing 253**

Integrated Test Facility (ITF)	254
Transaction Selection	256
Systems Control Audit Review File	256
On-Line Audit Monitor	259
Transaction Selection and Process Data Recording	261
Audit Indicators	261
Extended Records	262
Snapshots	262
Audit Processing Facilities Within Applications	263

Audit Modules	263
Audit Hooks	265
Summary	265

## **17. Audit Tools and Techniques for Database Management Systems 267**

The Auditor's Study and Evaluation of Internal Accounting Control	267
The Use of Database Management System Facilities in the Review of the System and in Testing	268
Data Dictionary/Directory	269
Backup and Recovery Facilities	269
Testing and Debugging Aids	270
<i>Facilities to Test Linkages and Pointers, 270 Trace Facilities, 272 Operation of the Database Management System in Test Mode, 272</i>	
Query Facilities	272
<i>Query Audit Capabilities, 272 Audit Reliance, 272</i>	
The Use of Generalized or Customized Audit Software With a Database Management System	273
Copying Relevant Records Onto a Sequential File	273
Modified Generalized Audit Software	273
Customized Audit Programs	273
Selecting Appropriate Audit Procedures	274
Summary	275

## **V. Other Data Processing Environments and EDP Auditing Activities**

### **18. Auditing Service-Center-Produced Records 279**

Operations of a Computer Data Processing Service Center	280
Responsibilities and Controls at the Service Center	280
User Controls	281
Auditor Concerns When a Service Center Is Used for Financial Records	282
Third-Party Reviews of the Service Center	283
Auditor Activities	284
Evaluation of User Controls	284
Evaluation of Service Center Controls	285
Evidential Matter	286
Time-sharing Computer Service Centers	286

## Contents

Basic Characteristics of Time-sharing	286
Control Features of Time-sharing	287
Summary	288

### **19. Control and Audit Considerations for Minicomputer and Microcomputer Systems 289**

Minicomputer and Microcomputer Definitions and Differences	289
Minicomputers and Distributed Data Processing	290
Effects of Minicomputers on EDP	291
Specialization	291
Hardware and Software Standards	291
Programming Languages	291
Logging and Audit Trails	291
File/Program Backup	291
Hardware Vendor Support	292
Audit Concerns and Audit Planning When Minicomputers Are Used for Financial Records	292
Audit Planning	292
Continuation Provisions with Minicomputers	292
Control and Evaluation in Minicomputer Systems	293
General Controls in the Use of Minicomputer Systems	293
Physical Security for Minicomputers	298
Application Controls in the Use of Minicomputer Systems	299
Performing Audit Tests	300
Control and Audit When Microcomputers Are Used for Data Processing	300
Summary	301

### **20. Audit Staff Selection, Organization, and Training for EDP Auditing 303**

Options in Organization of EDP Auditing Function in a CPA Firm	303
No Internal Specialization	304
Outside EDP Audit Specialists	304
EDP Audit Specialists on the Audit Staff	304
Management Advisory Services EDP Consultants Serving as EDP Audit Specialists	304
Technical Proficiency for Auditing Computer-Processed Records	305
Standards for Technical Proficiency	305
EDP Knowledge for All Auditors	305
Computer Audit Specialists	306
Academic Preparation for EDP Auditing	307



Sources of Continued Staff Training	308
Continuing Professional Education Courses Offered by the AICPA and State Societies	308
Courses Offered in the Business Environment	308
Courses by Colleges and Universities	308
Courses at Local Technical Schools	308
Seminars and Short Courses	308
Self-Instruction and Programmed Learning	309
In-Firm Courses	309

## **APPENDIXES**

<b>Appendix A. Questionnaire for the Evaluation of Internal Control in Computer Data Processing</b>	<b>313</b>
---	------------

<b>Appendix B. Flowcharting</b>	<b>345</b>
---------------------------------	------------

<b>Appendix C. An Example of Documentation</b>	<b>367</b>
--	------------

<b>Appendix D. Checklist for Evaluating the Audit Trail in a Computer Data Processing Application</b>	<b>389</b>
---	------------

<b>Appendix E. Audit Software Selection Checklist</b>	<b>395</b>
---	------------

<b>Appendix F. Evaluation of Controls over Processing Performed by a Data Processing Service Center</b>	<b>403</b>
---	------------

<b>Index</b>	<b>407</b>
--------------	------------



---

# Tables and Figures

## Chapter 2

**Table 2-1** Rough sequence for computer audit activities using computer audit programs or audit software and computer-stored client data, 25

**Figure 2-1** Decline in operational business activities for all industries following a complete data center failure (based on data collected in a survey of 36 companies), 16

## Chapter 3

**Figure 3-1** Flow of logic and activities in the auditor's study and evaluation of internal control in EDP systems, 28

## Chapter 4

**Table 4-1** Application development life cycle, 52

**Figure 4-1** Organization chart for a small data processing installation, 42

**4-2** Organization chart for a medium-sized data processing installation, 43

**4-3** Organization chart for a large data processing installation, 44

## Chapter 5

**Figure 5-1** Program change form, 63

**5-2** Data dictionary segment description, 65

**5-3** Data dictionary data element description, 66

**5-4** Data dictionary indexes, 68

**5-5** Computer-generated flowchart, 70

**5-6** Sample cross-reference report, 71

**5-7** Computer-generated file description, 73

**5-8** Programmer-written code using abbreviations, 74

**5-9** Code formatted by the formatting package, 76

**5-10** Master library index, 77

**5-11** IRS Revenue Procedure 64-12, 80

**5-12** IRS Revenue Procedure 71-20, 82

## Tables and Figures

### Chapter 6

- Figure 6-1** Odd parity coding for computer using eight-bit byte plus parity check bit, 86
- 6-2** Vacuum-column tape unit (right) and streaming tape unit (left). Courtesy of Control Data Corporation, 88
- 6-3** Magnetic tape data record, 89
- 6-4** Magnetic tape parity check, 90
- 6-5** Magnetic disk storage Courtesy of Control Data Corporation, 92
- 6-6** Waveform diagram, 93

### Chapter 7

- Figure 7-1** Internal memory allocation and problem of residual data, 101
- 7-2** Database setup, 107
- 7-3** Database use, 107

### Chapter 8

- Table 8-1** Coverage provided by different policies for data processing fire risks, 116
- 8-2** Computer room access, 119
- Figure 8-1** Internal magnetic tape labels – header and trailer records, 117
- 8-2** External magnetic tape label, 118
- 8-3** File protection ring for magnetic tape (write enable), 118
- 8-4** Librarian's log of program and file use, 121
- 8-5** Tape reel history card, 122
- 8-6** Grandfather, father, and son backup for magnetic tape files, 128
- 8-7** Backup and recovery for on-line disk files, 130

### Chapter 9

- Table 9-1** Processing methods in descending order of complexity, 143
- 9-2** Alternative designs in on-line entry/on-line processing, 144
- Figure 9-1** Two basic data processing cycles, 136
- 9-2** Immediate processing of all transactions, 145
- 9-3** Memo updating, 145
- 9-4** Immediate processing with periodic updating of file, 146
- 9-5** Immediate validation with access to master file, 147
- 9-6** Immediate validation without use of reference data, 148
- 9-7** File-order batch processing, 150

### Chapter 10

- Figure 10-1** Examples of output data that can aid in output review for reasonableness, 159

- 10-2** Generalized input data validation, 160
- 10-3** Circulating error file, 163

## Chapter 12

- Table 12-1** Needs for and sources of tracing information, 189
- Figure 12-1** Audit trail tracing, 184
- 12-2** Reference and control lists, 188
- 12-3** Transaction source references, 191
- 12-4** Audit trail in batch processing of hourly payroll, 194
- 12-5** Audit trail in on-line sales order entry, 195
- 12-6** Audit trail in accounts receivable aging, 196

## Chapter 13

- Figure 13-1** Access considerations for audit software, 203
- 13-2** Audit software criteria reports, 204
- 13-3** Audit software report using computation and comparison functions, 206
- 13-4** Audit software report with histograms, 208
- 13-5** Audit software using a coding approach, 209
- 13-6** Audit software using a checklist approach, 210
- 13-7** Precompilers versus interpreters, 212

## Chapter 14

- Figure 14-1** Specialized audit software – Instructions to sample an accounts receivable file, 223
- 14-2** Specialized audit software – Output from accounts receivable sample, 223
- 14-3** Flowchart tracing method for test data development, 226
- 14-4** Test data generator, 228
- 14-5** Test data generator output, 229
- 14-6** COBOL optimizer program execution profile, 230
- 14-7** COBOL optimizer report of unexecuted paragraphs, 230
- 14-8** Flowchart of parallel processing logic, 232

## Chapter 15

- Figure 15-1** SMF processing flowchart, 237
- 15-2** SMF reporting system, 238
- 15-3** Program execution report, 239
- 15-4** Peripheral usage report, 241
- 15-5** Audit processing of job accounting data, 242
- 15-6** Financial ratios from time-sharing service, 250

## Tables and Figures

- 15-7** Random number generation to produce random days for audit sampling, 251
- 15-8** Random number generation to produce random page and line numbers for audit sampling, 252

## Chapter 16

- Figure 16-1** Integrated test facility, 255
- 16-2** SCARF output report, 257
- 16-3** On-line audit monitor, 260
- 16-4** Audit indicator output report, 262
- 16-5** Snapshot output report, 264

## Chapter 17

- Figure 17-1** Forward and backward chaining, 271
- 17-2** Approaches to using generalized audit software in a database environment, 274

## Appendix B

- Figure B-1** Basic flowchart symbols, 346
- B-2** Standard flowchart symbols, 349
- B-3** Flowcharting conventions, 360
- B-4A** Data flowchart, 362
- B-4B** Control flowchart, 363
- B-5** Sales order processing horizontal flowchart, 364

## Appendix C

- Figure C-1** Program narrative, 369
- C-2** Program flowchart – Program HOUSEKEEPING SECTION, 372
- C-2** Program flowchart – Program MAINLINE SECTION, 373
- C-2** Program flowchart – Program WRITE-PAGE-HEADING SECTION, 375
- C-2** Program flowchart – Program OVERALL-CONTROL-TOTALS SECTION, 376
- C-2** Program flowchart – Program TEST-PART-KEY-NUMERIC SECTION, 377
- C-2** Program flowchart – Program DISC-I-O-ERROR SECTION, 378
- C-2** Program flowchart – Program SHUT-DOWN SECTION, 379
- C-3** Program source code listing – IDENTIFICATION and ENVIRONMENT DIVISIONS, 380
- C-3** Program source code listing – DATA DIVISION, 380
- C-3** Program source code listing – PROCEDURE DIVISION (HOUSEKEEPING and MAINLINE SECTIONS), 381
- C-3** Program source code listing – WRITE-PAGE-HEADING, OVERALL-CONTROL-TOTALS, and TEST-PART-KEY-NUMERIC SECTIONS, 382

## Tables and Figures

- C-3** Program source code listing – DISC-I-O-ERROR SECTION, 383
- C-3** Program source code listing – SHUT-DOWN SECTION, 383
- C-4** File description – Inventory transaction file, 384
- C-4** File description – Inventory master file, 385
- C-5** Error message list, 386
- C-6** Restart and recovery, 388





---

# Preface

The first edition of *Auditing & EDP*, published in 1968, was the result of the efforts of a special AICPA task force. Professor Gordon B. Davis, on leave from the University of Minnesota as computer consultant to the AICPA, was chairman of the task force and had primary responsibility for directing the work of the group, for drafting materials, and for editing the book.

The second edition updates the material, reflecting changes in computer technology and audit techniques that have occurred during the fifteen years since the original study. The revision is the joint effort of Professor Gordon B. Davis of the University of Minnesota, Donald L. Adams, AICPA vice president, administrative services (a member of the original task force), and Carol A. Schaller, manager in the computer services division of the AICPA during the project and now with Ernst & Whinney.

The book has been prepared to—

- 1 Guide CPAs in auditing the financial statements of organizations that use computers for record-keeping.
- 2 Provide source materials for training and professional development purposes.
- 3 Provide a discussion of alternative concepts and methods for auditing in the rapidly changing computer environment.

This book is intended for CPAs who have a fundamental understanding of EDP. It is not intended as a basic educational tool to teach data processing. Where needed for clarification, however, certain elements of EDP systems are explained.

The book has five logical sections plus appendixes. Section 1, chapters 1 through 3, examines auditing procedures followed when the client uses a computer to process financial records. It provides a background and framework for the sections that follow.

Section 2, consisting of chapters 4 through 9, describes general controls over the computer data processing functions. These controls include the organizational and management practices, documentation requirements, hardware controls, system software and database management software controls, and procedures for safeguarding availability, access, and use of data processing resources.

Section 3 defines controls that are specific to individual applications. This section consists of chapters 10 through 13 and includes a chapter on the different control implications of various data processing methods and a chapter on the audit trail in computer processing applications.

## Preface

Section 4 explains computer audit tools and techniques. Individual chapters cover generalized audit software systems (chapter 14), audit techniques for use after processing (chapters 15 and 16), audit techniques for use concurrent with processing (chapter 17), and techniques for auditing database management systems (chapter 18).

Section 5 explains special audit situations. Individual chapters cover audits in which service centers process the financial records (chapter 18), define the control and audit effects of minicomputer and microcomputer systems (chapter 19), and explain organization and training for computer auditing (chapter 20).

The appendixes provide a sample questionnaire for controls in computer data processing (Appendix A), a summary of flowcharting conventions (Appendix B), an example of suitable documentation (Appendix C), a checklist for evaluating the audit trail (Appendix D), an audit software selection checklist (Appendix E), and an evaluation of processing performed by a data processing service center (Appendix F).

The revision of *Auditing & EDP* has been reviewed by approximately two hundred members of the American Institute of Certified Public Accountants with knowledge and experience in the field of auditing and computer data processing. Their suggestions and criticisms have improved the book, and we wish to express our gratitude for their assistance. However, this book does not establish auditing standards for EDP, and it does not represent an official position of the AICPA or any of its senior technical bodies. Rather, it expresses the views of the authors, based on their experience and judgment.

---

**SECTION I**

# **The Audit and Computer Data Processing**

The three chapters of section I describe the background and framework for the sections that follow

Chapter 1—The Auditor and the Computer

Chapter 2—Planning and Applying Audit Procedures for Computer Data Processing

Chapter 3—The Study and Evaluation of Internal Control in Computer Data Processing

Chapter 1 surveys the impact of the computer on the processing of financial records and on data processing controls and summarizes the effect of the computer on the work of the auditor—on services provided, on the evaluation of internal accounting control, on specific audits, on the relationship of internal and external auditors, and on technical proficiency

Chapter 2 explains four control classifications used in describing EDP controls and then, using the concept of accounting and administrative control, defines the scope of the audit with respect to EDP. General and application controls are described in detail because they are basic to the study and evaluation of internal control in EDP systems. The chapter also outlines risk analysis of errors and irregularities, surveys the audit program for EDP, and explains some considerations that arise in planning and scheduling computer-related audit procedures

Chapter 3 explains the logical audit process for review and evaluation of internal accounting control for computer data processing and surveys the review and evaluation of provisions for continuation of computer data processing capabilities



---

# The Auditor and the Computer



The certified public accountant provides a variety of services related to computer data processing. This book is intended for the independent auditor of organizations having computer-based data processing systems, but the controls, control procedures, and audit procedures may also be useful to system designers, internal auditors, and other individuals involved in the design or testing of computer data processing systems or in related auditing activities.

## The Computer in Data Processing

The computer is one of the most significant technological developments of the twentieth century, and its importance has been discussed in both popular and technical publications. Although the first commercially available computer was installed in 1951 and the first financial data processing function was performed in 1954, the extensive use of computers in business did not begin until the early 1960s. Commercial data processing is now the most common use of computers.

Dramatic changes in computer technology have resulted in smaller computers, increased processing speed, increased storage capacity, and reduced cost. Computer processing capabilities are now readily available at relatively low cost through larger in-house computers, small in-office computers, or through remote access to computers at another location.

### TRENDS IN APPLICATION OF COMPUTERS BY ORGANIZATIONS

The changes in hardware and software technology are reflected in information processing systems through increased use of the following features:

- Terminals having logic and storage capabilities, called "intelligent terminals," for transaction data entry and transaction output
- Data communication facilities for communication between computers or between terminals and a computer
- Small computers (minicomputers and microcomputers)
- Database management systems and a data administration function

## 4 The Auditor and the Computer

These four features of computer system technology are reflected in system design trends toward—

- On-line, real-time systems
- Distributed data processing systems
- Systems with databases and database management systems

On-line systems provide for real-time (immediate) entry of transactions, retrieval of data, and updating of data in files. These systems use terminals and usually use data communication facilities. The on-line systems allow individual transactions to be processed when they occur, thus supporting many organizational processes that rely on individual transactions and prompt processing.

There is a tendency in information processing systems to distribute processing capabilities to user locations rather than to concentrate them in a central processing center. Distributed processing can take many forms, depending on which capabilities are dispersed. The following are some examples of distributed processing systems:

- Local, independent computers in several locations
- Local computers that are part of a network controlled by a central computer
- Remote job entry stations that process input of data and handle output
- Small computers dedicated to a few uses or a few users

Databases and database management software may be used with traditional batch processing, with on-line systems, and with distributed systems. Database systems allow the organization to treat data as an organizational resource, separating data and data management from the programs that create, update, or use the data. The database concept is implemented by database management system software and by an organizational position, the database administrator.

## The Computer and Client Services by CPAs

The computer has affected all aspects of the practice of public accounting, including audit, tax, accounting, and management advisory services, in firms of all sizes.

### AUDIT PRACTICE AND THE COMPUTER

If a client uses a computer to process significant accounting data relevant to the audit, the study and evaluation of internal accounting control must include a review of the computer processing system. A preliminary phase of the review is performed to plan the audit procedures to be applied. The review process must be completed by more detailed review activities if the auditor plans to rely on the computer data processing controls. Moreover, the auditor may use the computer in the audit to obtain data, perform analyses, test to evaluate data, select samples for substantive tests, or compare evidential matter obtained from physical counts.

### TAX PRACTICE AND THE COMPUTER

Computer software is available from commercial vendors and from some CPA firms to provide support for tax research and tax planning and to prepare tax returns. Clients

have access to these services through a time-sharing terminal, a remote batch input/output device in the CPA office, an in-house computer or microcomputer operated by the CPA firm, or by mailing input forms to a service center

### **ACCOUNTING SERVICES AND THE COMPUTER**

The computer is cost effective in processing accounting transactions, producing transaction documents, and preparing reports. Some CPAs provide these services through an in-house computer center, other CPAs provide auxiliary services for the preparation of data and the analysis of output but use a data processing service center to process client data. Also, CPAs may be asked to advise clients on selection and use of a service center.

### **MANAGEMENT ADVISORY SERVICES**

A substantial segment of management advisory services provided by CPA firms is related to the use of the computer. Examples of such services are as follows:

- Preparation of an information system master development plan
- Recommendations in the selection of hardware, software, or processing services
- Computer system performance evaluation
- Design or evaluation of EDP organization
- Evaluation of EDP control
- Review of EDP security

### **ACCOUNTING PRACTICE RESEARCH**

An activity related to audit practice is computer-aided accounting research. There are a variety of information retrieval services that search journals, government reports, and doctoral dissertations. Such searches are useful in obtaining analyses and research related to an accounting problem. The AICPA provides NAARS (National Automated Accounting Research System), a unique retrieval service that performs computer searches of financial statements, notes, auditors' reports, and accounting rules and regulations. The search process is used to research such matters as alternative accounting and reporting practices and the precedents for them.

## **The Effect of the Computer on Data Processing Controls**

An organization's computer data processing system consists of the following elements:

- Computer hardware
- System software (operating systems, compilers, and so on)
- Generalized utility software (programs that handle common tasks such as sorting)
- Personnel (operators, programmers, analysts, and so on)
- General operating procedures and controls
- Applications (specific programs and procedures for each application)
- Data (organized as files used by single applications or as databases used by several applications)

## 6 The Auditor and the Computer

The basic unit in a data processing system is an "application" or "application system," such as payroll, accounts receivable, or order entry. The processing procedures, operations, and management of an installation are either general, applying to many applications, or specific, applying to a single application. Stored data, also, may be managed by a database management system and available to many applications or may be specific to an application and controlled by the procedures for that application.

The installation of a computer produces different control elements and changes control procedures in the system. The changes may be classified as follows:

- Different organizational control procedures to direct and supervise computer data processing
- Different processing controls necessary because of the automation of processing for applications
- Controls in the computer system that substitute for controls in manual systems, which were based on human judgment and segregation of functions

### **ORGANIZATIONAL CONTROL PROCEDURES**

Among the control procedures for directing and supervising computer data processing is an information system master development plan, which is generally not necessary in a manual system. New positions, such as librarian, data administrator, and control clerk, perform the different control activities. Other examples of organizational controls are the standards and procedures for selecting, developing, testing, and implementing new applications.

### **PROCESSING CONTROLS NECESSARY**

A number of new processing controls are necessary when a computer is used in data processing. Hardware controls detect equipment malfunctions. Because computer data processing requires the preparation and input of data in machine-readable form, controls are needed to ensure that procedures for data preparation and input provide complete, correct, and valid data. Examples of data preparation and input controls are verification of data conversion, programmed data validation routines, and batch controls. Computer data processing methods may allow increased access to data files and databases; therefore, access and security codes, records of access, and other data access controls become important considerations.

### **COMPUTER SYSTEM ALTERNATIVES TO MANUAL CONTROLS**

In a manual system, internal accounting control to prevent or detect errors and irregularities or fraud relies heavily on organizational factors, such as alertness, judgment, acceptance of responsibility, and segregation of functions. Computer data processing reduces the number of persons directly involved in data processing. This concentration of responsibility reduces the controls based on segregation of functions. Automation also reduces the extent of control based on human judgment and alertness.

The computer often provides alternative controls that fulfill the same objectives. Indeed, some computer controls can be more effective than manual controls. For example, the computer may be programmed to test a data item for validity or reasonableness, using a specified set of criteria. Once in the program, the test will be executed



precisely as intended, without the loss of effectiveness that sometimes occurs in manual processing

## **The Study and Evaluation of Internal Accounting Control When a Computer Is Used**

The study and evaluation of internal accounting control when a computer is used in data processing is described in SAS 3, *The Effects of EDP on the Auditor's Study and Evaluation of Internal Control*, which is supported by the AICPA Audit and Accounting Guide, *The Auditor's Study and Evaluation of Internal Control in EDP Systems*. The methods defined by the SAS and the audit guide are described in chapter 3, but the essential features of the approach are summarized here.

The two types of controls associated with computer processing are

(a) general controls governing the operation and management of computer processing activities and (b) application controls for individual processing applications. Examples of general controls are division of duties, such control functions as database administrator, and processing controls. Examples of application controls are batch controls, check digits, and input error reports.

The study and evaluation of internal accounting control can be viewed as consisting of four parts—two review phases, tests of compliance, and evaluation of the system. The division of the review into two phases is an important logical separation. The preliminary phase provides an overall understanding of the basic structure and the extent to which EDP is used, an overall understanding of the general controls, an overview of the flow of application processing, and a preliminary understanding of controls for significant accounting applications. The completion of the review expands the work of the preliminary phase with more detailed review activities. Its purpose is to gather additional information to design tests of controls on which reliance is to be placed.

The preliminary phase of the review is necessary whenever a computer is used to process accounting records. It provides a basis for overall judgments regarding the role of computer processing in records preparation and the effect of computer processing on each significant accounting application. Thus, it provides a basis for planning the remainder of the audit.

On the basis of these preliminary judgments, the auditor decides whether to rely on the data processing controls for each significant application. A decision not to rely on data processing controls for an application eliminates the need for any further evaluation of application controls or any further evaluation of general controls, unless an evaluation is required for other applications. The auditor then relies on review and testing of manual controls and substantive tests to provide sufficient evidential matter. On the other hand, a tentative decision to rely on data processing controls for an application results in the completion of the review with more detailed review procedures. If, after applying the detailed procedures, the auditor again decides to rely on the data processing controls, compliance tests are made. Compliance testing is followed by appropriate evaluation of user controls and substantive testing.

The preliminary and completion phases of the review are separate steps in the audit process. However, in some audits it may be more efficient to perform both phases at the same time. Also, after the initial audit of a new client or after a new or revised computer system has been evaluated, the preliminary phase of the review may be concerned mainly with computer system changes and their effects on the audit plan.

## 8 The Auditor and the Computer

### **Special Reviews of Computer Data Processing**

The study and evaluation of internal accounting control made as part of an audit provides information that is usually of interest to management. The auditor will normally submit observations and recommendations in a management letter. However, the management may wish the auditor or management advisory services personnel to perform more complete reviews for management purposes, which may include the following

- Data processing management review
- Application postimplementation review
- Computer processing continuation review
- Privacy compliance review

#### **DATA PROCESSING MANAGEMENT REVIEW**

The CPA may be asked to review the organization and management of the computer data processing function. The reviewer evaluates the completeness and appropriateness of management policies and procedures and tests compliance with these policies and procedures. A complete review might cover the following areas

- Organization structure
- Staffing and training
- Management planning and control
- Application development and maintenance procedures
- Hardware and software acquisition procedures
- Use of current hardware and software
- Operational procedures

#### **APPLICATION POSTIMPLEMENTATION REVIEW**

A review of each computer data processing application should be performed after the application has been implemented (perhaps after one year). The postimplementation review team normally includes representatives from the development staff, the users, and internal or independent auditors. The team investigates the use of the application, compares the expected benefit with the actual benefit, and assesses the development process. The evaluation report makes recommendations regarding the application and, perhaps, management and control processes for future application development.

#### **COMPUTER PROCESSING CONTINUATION PROVISIONS REVIEW**

The concentration of equipment and data files in a computer facility increases the potential for financial loss from the interruption or loss of data processing capabilities. To reduce this potential for loss, an installation should have adequate protection against such risks as fire, flood, wind, and riot. There should be control over access to (1) the installation itself, (2) files, and (3) use of the hardware or software. There should be a backup and recovery plan and procedures for maintaining backup copies of files, software, procedures manuals, forms, and so on. Management is responsible for implementing EDP continuation provisions and for supervising and reviewing their operation and effectiveness. The review and testing of these provisions can be performed during the audit or as a special review.

## PRIVACY COMPLIANCE REVIEW

Privacy legislation has been directed primarily at governmental units and some specialized business organizations, such as credit bureaus. There appears to be a trend toward limiting the personal data that can be collected and held in files and limiting the alternate use of data obtained for legitimate purposes. Special reviews are useful for evaluating compliance with privacy legislation and management policy regarding use of personal data.

## Relationship of Independent Auditors to Internal Auditors

The involvement of the internal auditors of an organization in the computer data processing area may take the following different forms:

- No EDP audit activity by internal auditors
- Some EDP audit activity by internal audit staff but no specialization
- Specialization in EDP auditing by one or two internal auditors
- Separate EDP audit staff with several internal auditors specializing in EDP auditing

There is a trend toward specialization among audit groups in large organizations, this specialization is particularly evident in the establishment of separate EDP audit groups.

The continuous involvement of internal EDP auditors with the computer installations in a single organization allows in-depth understanding of the installation. The internal EDP auditor is also able to repeat tests at frequent intervals. The following are some activities that might routinely be included in the scope of internal EDP auditing but are not within the usual scope of the audit of financial statements:

- Review of control requirements for a new application during the design phase of development
- Postimplementation review of applications for use, effectiveness, and efficiency
- Evaluation of operating efficiency of data processing facilities
- Review and compliance tests for computer data processing continuation provisions, including facility protection and backup and recovery
- Use of audit techniques for continuous testing or monitoring of applications

The principle of independent auditor reliance on the work of internal auditors is well established. (See SAS 9.) This relationship can be especially beneficial in the case of EDP auditing because of the opportunity that internal EDP auditors have to monitor changes in data processing that would be difficult for an independent auditor to review and test.

## Auditors' Need for Competence in Computer Data Processing

If an audit involves computer-maintained accounting records, the independent auditor must have sufficient competence in EDP systems to conduct the audit properly. This requirement follows from the general requirement for "adequate technical training and proficiency as an auditor," defined by generally accepted auditing standards and rules of conduct. The first general standard states, "The examination is to be performed by a person or persons having adequate technical training and proficiency as an auditor."

## 10 The Auditor and the Computer

Rule 201 of the AICPA Code of Professional Ethics also recognizes the importance of competence, requiring that "a member shall not undertake any engagement which he or his firm cannot reasonably expect to complete with professional competence "

In general, the audit of financial statements prepared from records processed by a computer-based system requires the auditor to possess a basic understanding of computers, computer facility organization, computer data processing methods, computer processing controls, and computer-assisted audit techniques. The complexity of the audit activities varies with the complexity of the processing system, thus, the extent of required computer proficiency varies, too. For example, an audit for a company having a small, batch-oriented data processing installation requires less proficiency than an audit of a company having a large, complex, on-line computer system. The availability of visible audit evidence that may be subjected to compliance testing or substantive testing and the planned audit approach also affect the auditor's need for proficiency in computers.

Not all members of the audit team need be equally competent in computers.

The following approaches are used by CPA firms in providing sufficient expertise to the audit team:

- General staff development in EDP skills
- EDP audit specialist assigned to each audit requiring EDP skills
- Management services computer specialist assigned to the audit team to assist if EDP skills are required

Chapter 20 surveys methods for selecting computer audit staff and obtaining and maintaining adequate technical skills in computer processing.

### Summary

The trend in computer technology continues to be toward smaller, more powerful, less expensive equipment. The trend in organizational information processing systems is toward terminals, data communication facilities, distributed processing facilities, and database management systems. These developments affect audit practice, as well as other CPA services. Computerization confronts the auditor with organizational control procedures to direct and supervise computer data processing, automated processing and data storage procedures, and unique computer system controls.

The study and evaluation of internal accounting control when a computer is used in data processing involves an understanding of both general controls and application controls specific to the accounting applications having audit significance. The review of internal accounting control begins with a preliminary phase, the results of which are used in planning the remainder of the audit. If the auditor expects to rely on the controls in computer data processing, the review process is completed by a more detailed review phase.

Special reviews that may be performed in the computer data processing organization include management reviews, application postimplementation reviews, continuation provisions reviews, and privacy compliance reviews.

The growth of computer data processing in organizations has affected not only the independent auditor but also internal auditors. There is a trend toward use of internal EDP audit specialists who are involved on a fairly continuous basis in evaluating computer operations, applications, and security within an organization.

---

# Planning and Applying Audit Procedures for Computer Data Processing

# 2

No separate set of audit procedures exists for computer data processing. Instead, the audit activities related to computer processing environments are an integral part of the total activities in the financial audit. Therefore, in planning and performing audit procedures for an EDP function, the auditor should follow the third standard of field work, contained in SAS 1, section 150.02, which states, "Sufficient competent evidential matter is to be obtained through inspection, observation, inquiries, and confirmations to afford a reasonable basis for an opinion regarding the financial statements under examination." However, because unique characteristics are associated with this portion of the audit, it is useful to make it an object of special study and specific training.

This chapter discusses various aspects of controls and clarifies their use. Two broad classifications of controls are explained in detail: accounting controls/administrative controls and general controls/application controls.

The audit plan should meet audit objectives with respect to EDP, and, accordingly, audit procedures should reflect the unique characteristics associated with the EDP environment. In developing the audit plan for the EDP area, the auditor must consider special problems in planning and scheduling audit-related procedures. This chapter provides an overview of audit planning to be applied to computer data processing activities and comments on the planning and scheduling of these EDP-related audit activities.

## Classification of Controls in Computer Data Processing

In discussing the study and evaluation of internal accounting control and other audit and control procedures related to computer data processing, several different classifications of controls are used. The same control may be classified in more than one control category. This book uses four classifications, as follows.

- 1 *Accounting controls or administrative controls* Accounting controls are concerned mainly with safeguarding of assets and the reliability of the financial records, while administrative controls are concerned mainly with operational efficiency. As is explained later in the chapter, this classification is used to define the scope of the audit, in general, accounting controls are within the scope of the auditor's study and evaluation, and administrative controls are not.

## 12 Planning and Applying Audit Procedures for Computer Data Processing

- 2 *General controls or application controls* General controls are the organization, management, and operation controls within which applications are developed, maintained, and executed. Application controls are the specific controls included in the computer or manual procedures of an application. The classification is used in the study and evaluation of internal accounting control to separate the study of controls that apply to the environment for many applications from the study of controls specific to each application within the scope of the audit.
- 3 *Development, processing, or continuation controls* Development controls ensure the development of applications that process information according to specifications and that have adequate controls and audit trails. Processing controls ensure the complete, accurate processing of valid transactions using the appropriate files and program versions. Continuation controls ensure the availability of computer operations capabilities by protecting against loss or damage and by providing backup and recovery. This classification is used to clarify the purpose of controls for the three major activities of (1) developing and maintaining applications, (2) processing applications, and (3) providing for continuation of computer processing capabilities.
- 4 *Preventive, detective, or corrective controls* This classification identifies the timing of the control in the work flow. It is used to determine whether a control acts primarily to prevent an error or irregularity, to detect one after it occurs, or to correct it. This classification is useful in the analysis of controls for a specific transaction.

Each of these classifications has a purpose in the evaluation of computer data processing control. The auditor first classifies controls as being accounting controls or administrative controls, the accounting controls within EDP are then classified as general EDP controls and EDP application controls. The general and application controls may be analyzed further in regard to development, processing, or continuation or in regard to whether they are preventive, detective, or corrective. Three examples illustrate this method of classification of an EDP control.

- 1 *Input validation test performed by application programs to detect data items outside of the normal range of values*
  - Accounting control because it is directed at the reliability of the financial records
  - Application control because it is part of an application
  - Processing control because it applies to the processing of data for an application
  - Detective control because it detects an error that has already occurred in recording or entering the data item
- 2 *Off-site storage of transaction files and master files*
  - Administrative control because it is concerned with the recovery of files lost through processing, destructive events, or improper activities. It is not required for the reliability of the financial records or for safeguarding assets. (Even though some may consider the data files an asset, they are not recorded as an asset in the financial records.)
  - General control because the procedure applies to files from many applications
  - Continuation control because the purpose is backup and recovery
  - Corrective control because it is used to correct the error by providing data for recovery procedures

- 3 *Use of operator sign-on procedures before transactions may be entered at a terminal*
- Accounting control because the procedure is designed to prevent unauthorized entry of transactions. It is also an administrative control if it is used to obtain efficiency data on operator performance.
  - Application control if the procedure is designed as a single application. A general control if there is an access control program that handles access for all applications.
  - Processing control because it applies to current processing.
  - Preventive control because it prevents entry of transactions by unauthorized personnel.

## General Scope for Audit Procedures in an EDP Environment

Internal control has been subdivided in the auditing literature into accounting controls and administrative controls to clarify the scope of the study and evaluation contemplated by generally accepted auditing standards. The classification is also used to clarify the role of the auditor in the review of provisions to ensure continuation of computer data processing capabilities.

### ACCOUNTING CONTROL: WITHIN THE REQUIRED SCOPE

Accounting control and administrative control are conceptually distinct, but in practice many procedures are used for both purposes. A procedure or activity used for both accounting control and administrative control is within the scope of the evaluation of internal control.

SAS 1, section 320, also identifies basic concepts and design characteristics included within the scope of accounting control.

#### Basic Concepts of Accounting Control

##### *Segregation of functions*

No person is in a position that enables him "both to perpetrate and to conceal errors or irregularities in the normal course of duties."

#### Corresponding EDP Accounting Control

##### *Segregation of EDP functions*

- Segregation of functions in the development of computer applications, custody of programs and data files, and operation of applications.
- Segregation of responsibilities, so that the person who authorizes changes in application logic or in stored data does not make those changes.
- Divisions of control duties among users, data processing personnel, and control personnel.

(continued)

## 14 Planning and Applying Audit Procedures for Computer Data Processing

### *Basic Concepts of Accounting Control (cont )*

---

#### *Execution and authorization of transactions*

"Transactions are executed in accordance with management's general or specific authorization "

#### *Recording of transactions*

"Transactions are recorded as necessary to (1) permit preparation of financial statements and (2) to maintain accountability for assets "

#### *Access to assets*

"Access to assets is permitted only in accordance with management's authorization "

#### *Comparison of recorded accountability with assets*

"The recorded accountability for assets is compared with existing assets at reasonable intervals "

### *Corresponding EDP Accounting Control (cont )*

---

#### *EDP controls to ensure that only authorized transactions are processed using authorized computer programs*

- Controls to check the authorization of an individual entering a transaction and to test the transaction against criteria for authorized transactions
- Controls over development, testing, custody, and maintenance of computer programs and related documentation to ensure that program logic meets management's specifications

#### *EDP controls to ensure complete, correct, and timely recording and processing of transactions*

- General and application controls to prevent or detect loss, errors, or omissions in recording, preparing, and entering data for processing
- Controls to validate data, to reject erroneous records, and to control timely correction and re-entry of input
- Controls to ensure availability and use of correct files and programs

#### *EDP access control*

- Controls over access to the computer data processing resources, including equipment, software, and files
- Access controls over use of terminals to initiate and process transactions involving assets or records of assets
- Controls over preparation and issuance of documents that authorize disbursement of assets (such as checks and shipping orders)

#### *Periodic comparison of stored data records with assets*

Establishment of regular procedures to compare the computer-based records with physical counts, records from parties outside the organization, and other evidence of assets



Controls not meeting these criteria are administrative

*Administrative control* includes, but is not limited to, the plan of organization and procedures and records that are concerned with the decision processes leading to management's authorization of transactions [SAS 1, section 320.27]

Although this definition of administrative control is not complete, it helps to distinguish between the two types of controls. Administrative controls ensure that useful information is provided and used in achieving the organization's objectives. Controls to ensure efficiency in operations are also part of administrative control. An auditor may accept an engagement to study and evaluate administrative control, but it is not ordinarily a part of an audit of financial statements.

### **COMPUTER OPERATIONS CONTINUATION PROVISIONS: AN OPTIONAL EXTENSION OF SCOPE**

Continuation provisions are controls, procedures, and arrangements to provide reasonable assurance that an organization will be able to recover from loss or destruction of data processing facilities, hardware, software, or data and be able to continue to provide accurate, complete data for operations, management decisions, and financial statements. Continuation controls and provisions can be considered to be within administrative rather than accounting controls. The auditor may choose to review these controls and provisions, especially if the data and records are vital to continuation of the business in the event of computer failure or destruction.

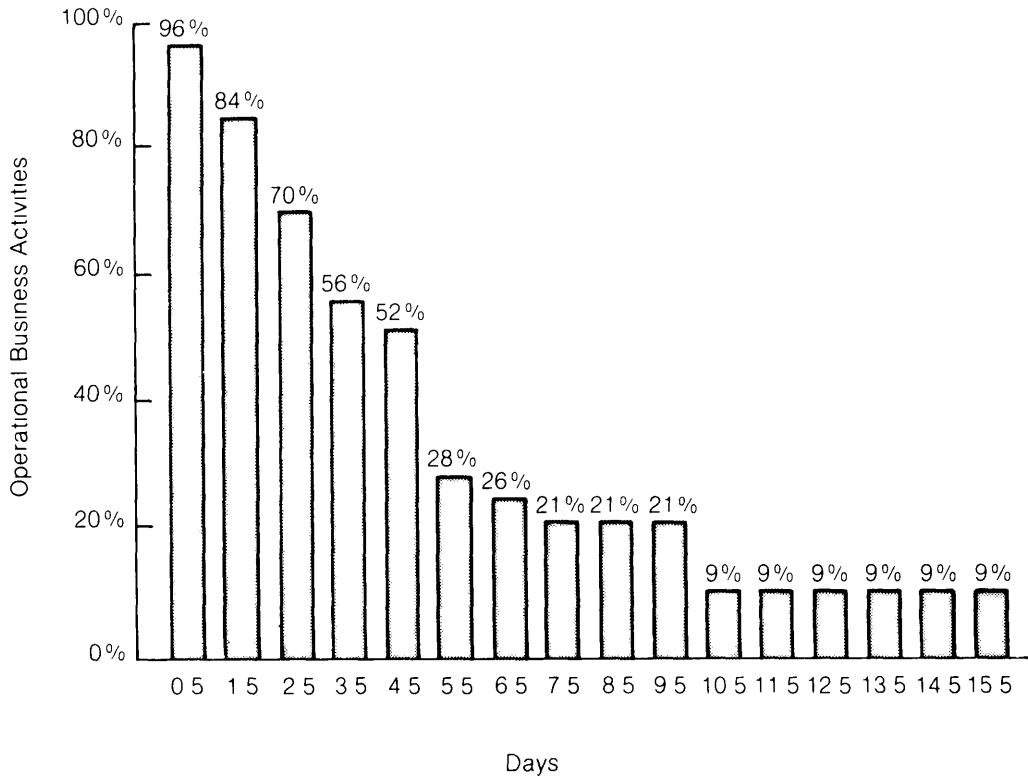
Continuation provisions for data processing operations include physical protection and backup and recovery. Physical protection includes physical access security, fire protection, wind and water protection, and electrical service provisions. Backup and recovery provisions include retention of copies of data files and copies of software, arrangements for access to backup hardware and other facilities on short notice, provisions for supplies of forms and other media, tested recovery plans, and insurance.

An organization's lack of adequate backup and recovery procedures does not affect an auditor's opinion on financial statements for periods already ended. However, continuation of the computer data processing system may be vital to the continuation of the organization itself. Failure of processing capabilities, inability to access records, or destruction of records may cause large losses and may impair the organization's ability to survive. In a 1978 survey (figure 2-1), the thirty-six organizations responding estimated that only about one-half of their business activities would be able to continue for four and one-half days after computer processing ceased and only 9 percent could continue more than ten days without computer data processing.<sup>1</sup>

Management is responsible for maintaining the accounting system. This responsibility includes evaluating the risk of destruction or impairment of computer data processing capabilities and minimizing such risks by applying various continuation provisions. Management should ensure that continuation provisions are periodically reviewed by internal or independent auditors for effectiveness and efficiency.

---

<sup>1</sup> D. O. Aasgaard, P. R. Chenng, B. J. Hubert, and M. C. Simpson, *An Evaluation of Data Processing Machine Room Loss and Selected Recovery Strategies*, Working Paper 79-04, The Management Information Systems Research Center, University of Minnesota, 1978.



**Figure 2-1** Decline in operational business activities for all industries following a complete data center failure (based on data collected in a survey of 36 companies)

Independent auditors can include it as part of the study and evaluation of internal control or as a special review, however, it is generally efficient to integrate the continuation provisions review into the audit plan. Because of its importance to management and to others in the organization, many independent auditors and their clients have agreed to include continuation review in the procedures performed for the financial audit.

## General Controls and Application Controls

The auditor considers general controls and application controls in studying and evaluating internal accounting control in computer data processing systems.

### GENERAL CONTROLS

General controls are used in the development of applications and for computer processing activities that apply to several applications. They provide the control environment in which applications (such as payroll, accounts receivable, and inventory accounting) are processed. Specific application controls are used to control the flow of

transactions through the application. The following classification of general controls in the computer data processing system corresponds to the classification in the AICPA Audit and Accounting Guide, *The Auditor's Study and Evaluation of Internal Control in EDP Systems*

- Organizational and operational controls
- Application system development and maintenance controls
- Hardware and system software controls
- Access controls
- Procedural controls

Each of these general controls is described briefly, and additional explanation of the design and use of the controls is given in chapters 4 through 9

Organizational and operational controls are provided by the organizational plan and designation of duties and include the following

- Segregation of functions between the EDP department and users
- Control over authorization of transactions
- Segregation of functions within the EDP department

These controls are described in chapter 4

Application system development and maintenance controls are applied to the process by which new applications are developed and installed, to the process by which applications are changed and updated, and to the documentation procedures required to provide backup and application system maintenance. These controls are described in chapters 4 and 5.

Hardware controls can detect hardware errors or failures and in certain cases can correct them automatically, however, some controls are effective only if certain software procedures are followed. Examples of hardware controls are circuitry to detect and correct parity errors in storage, controls to detect data transmission errors, and controls to detect card read errors. Hardware controls are described in chapter 6

System software may contain control features to prevent operational errors. Examples include record counts and creation and checking of file labels. System software controls, including database software controls, are described in chapter 7

Access controls protect against unauthorized or improper use of computer data processing equipment and stored data. Organizational and operational access controls include the following

- Authorization for use of computer hardware, schedules for use, and the logging and reporting of actual usage
- Librarian function to control access to programs and data files (The procedures may include computer program management software)
- Data access control through a data administration function for data stored in a database
- Documentation access control
- Security codes and access codes (passwords) to limit access from remote terminals to persons authorized to enter transactions

Access controls are described in chapter 8

## 18 Planning and Applying Audit Procedures for Computer Data Processing

Procedural controls are specific procedures used to control operations and to ensure accurate and complete processing. They include the following:

- Control function to record control totals for input data and compare output totals with control totals
- Error control procedures to log errors and record corrections plus procedures to follow up uncorrected errors
- Procedures to examine output for errors and to distribute output to authorized recipients
- Written computer operation procedures for all operational steps, including procedures for handling errors, operational failures, and so on
- Operational provisions for backup and recovery in case of errors, failures, and so on
- Physical security procedures

Physical security and backup and recovery provisions are described in chapter 8, and general operational procedures are described in chapter 9.

### APPLICATION CONTROLS

Application controls are specific to a particular application. There are similarities between controls for different applications, but, for example, the specific controls for a payroll application will differ somewhat from the controls for an accounts receivable application. Application controls are described in chapters 10 and 11.

Application controls are generally classified as input, processing, and output controls. However, some application controls or control system activities, such as controls primarily for backup and recovery, may not fit into this classification, examples are transaction logs and before- and after-image logs.

Input controls ensure that data items to be processed are authorized, complete, and correct. "Input" refers to all the procedures used to submit data for processing and can include—

- Data capture and recording on a document or entry of data through a terminal or data collection device
- Batching of transactions
- Data conversion, such as keying from a document
- Data validation procedures applied to either individual transactions or batches of transactions
- Data correction procedures at time of entry for individual transactions or after periodic input processing of batched transactions

The input controls should detect lost or duplicated transactions, as well as transactions entered incorrectly. There should be controls not only to detect errors but also to ensure correction and prevent resubmission of erroneous data.

Processing controls provide reasonable assurance that all transactions have been processed and that the application processing has been performed correctly, using the correct file data, operator procedures, and processing logic.

Output controls provide reasonable assurance that the output is complete and correct and that it is distributed only to authorized users.

## Risk and Exposure Analysis in Audit Planning

There is a risk of material error in audited financial statements. This risk is a combination of two individual risks, according to SAS 39, paragraph 8,

the risk that material errors will occur in the accounting process used to develop the financial statements and the risk that any material errors that occur will not be detected by the auditor

The auditor may rely on internal accounting control to reduce the risk of material errors in the accounting process and on substantive tests to reduce the risk that material errors will go undetected.

Since not all accounting processes, accounts, and assets are equal in importance to the financial statements or equally susceptible to material error, audit resources are used most effectively if they are applied selectively. In other words, an auditor normally examines only those items that are important and susceptible to material error, not all assets or all transactions.

This concept is applicable to computer data processing. Not all computer processing activities and applications need be included in the scope of the audit—only those that are relevant to the processing of significant financial data for the financial statements. Audit procedures should reflect the importance of the data to the financial statements and the susceptibility of the activity or application to errors and irregularities that will result in material error in the financial statements. SAS 30, *Reporting on Internal Control*, paragraph 14, recommends that, in planning the scope of the engagement, the auditor should consider the nature of the entity's operations, including the susceptibility of assets to unauthorized use or misappropriation, the nature and volume of transactions, and the relative significance of the various classes of transactions and related assets.

Risk assessment for EDP should not be done in isolation, because computer data processing procedures are preceded and followed by other organizational activities and manual procedures.

The auditor must identify material risks and exposures (1) involved in the use of development resources and the use of hardware, software, and other facilities, (2) relative to applications that process financial data, (3) for transactions within an application, and (4) for data elements in transactions and stored data.

### EDP FUNCTION RISK AND EXPOSURE ANALYSIS

Most risks and exposures from errors and irregularities in the EDP function can be associated with specific applications, however, the use of EDP resources may contain direct errors and irregularities associated with no specific application.

#### Example of Error or Irregularity

Unauthorized or unrecorded disposition of hardware or software  
Unrecorded liability in connection with the purchase or lease of hardware or software

#### Effect of Error or Irregularity

Assets are overstated  
Liabilities are understated

(continued)

## 20 Planning and Applying Audit Procedures for Computer Data Processing

### Example of Error or Irregularity (cont.)

Use of hardware and software for unauthorized purposes

Unauthorized development activities using personnel, supplies, and facilities

Unauthorized copying of data and programs for private use or fraudulent disposition

### Effect of Error or Irregularity (cont.)

The financial statements are correct, but some expenses are not related to the enterprise's objectives

The financial statements are correct, but expenses exceed authorization

This has no direct effect on the financial statements, however, there may be serious future effects from, for example, disclosures to competitors

Some unauthorized use of facilities may involve relatively little expense and, therefore, will not be of material concern to the independent auditor. Management, however, may wish to prevent or detect such use in order to avoid an organizational climate conducive to fraud or other irregularity. For example, use of the facilities for keeping bowling team scores does not result in a major expense, but management may wish to either explicitly prohibit such use or have a procedure to authorize such minor, community-related use.

In addition to errors and irregularities affecting past financial statements, there are risks that errors and irregularities will destroy (or fail to protect) data processing capabilities. This may affect the future operations of the organization.

### **APPLICATION LEVEL RISK AND EXPOSURE ANALYSIS**

The auditor should also analyze the risk associated with sets of activities and transactions that make up a data processing application. As discussed earlier, the auditor must evaluate the significance of an application, including the importance of the data to the financial statements and the susceptibility of the assets to misappropriation. In different organizations, similar applications differ in their effect on the financial statements. For example, the accounts receivable application for a business that sells mainly for cash may have a minor effect on the financial statements, but the same application will be significant for an organization that sells mainly on credit.

### **TRANSACTION LEVEL RISK AND EXPOSURE ANALYSIS**

The exposure analysis at the application level encompasses sets of transactions associated with the application. A more detailed analysis can be performed on each significant transaction, including the risks of invalid, unauthorized, or unrecorded transactions, improper valuation, improper classification, improper timing, and the failure to record transactions in subsidiary records. The auditor assesses the probability that the risks have occurred, the probable frequency of occurrence, and the probable consequences.

The auditor should also consider the relative risk from error or irregularity associated with different types of transactions. Normally, the effect of an error or irregularity is limited to a single transaction, but the effect is more complex if the transaction triggers other transactions. An error or irregularity involving the maintenance or initial entry of stored data will affect all subsequent processing using the incorrect data, while an error or irregularity involving an inquiry will have no direct effect on the financial records. Transactions involving high dollar amounts entail greater risk than transactions

involving low dollar amounts. Finally, errors or irregularities in the processing of some applications are more likely to be detected and reported by personnel outside of data processing (such as customers, vendors, and shop personnel) than those for other applications.

### **DATA ITEM ANALYSIS**

Each transaction and each stored data record is composed of data items, and it may be desirable to extend the risk and exposure analysis to the data item level. Not all data items are equally significant to the financial statements, nor are all data items equally susceptible to error. Also, the trend to databases used by many applications means that the risk associated with data items in the database must be assessed separately from a single application or a single transaction. Among the auditor's considerations are whether the data items are used in applications that materially affect the financial statements, the origin of the data, the sources of updating, and the extent of review and quality assurance for the items.

## **Overview of the Audit Program as It Relates to Computer Data Processing**

The audit procedures for computer data processing relate directly to general audit objectives. On the basis of the audit objectives and the unique characteristics of the computer and the computer applications, an audit program may be prepared for audit activities related to electronic data processing.

### **UNIQUE CHARACTERISTICS OF AUDITS IN AN EDP ENVIRONMENT**

The following characteristics of the EDP environment affect the audit procedure:

- Computer data processing technology (hardware, software, and data communication)
- Technology-oriented procedures for developing and revising data processing application systems
- Frequent changes in computer data processing applications to reflect changing user requirements and changing computer technology
- Data records stored only on magnetic media, so that accounting records can be accessed and processed, by using the computer
- The availability of computer-assisted audit techniques

These characteristics necessitate an audit plan that takes into account the design, development, and maintenance of the computer system being used to process financial records.

### **STEPS IN THE AUDIT PROGRAM RELATED TO COMPUTER DATA PROCESSING**

Computer data processing affects the audit of financial statements at two important points. The first is the study and evaluation of existing internal accounting control,

## 22 Planning and Applying Audit Procedures for Computer Data Processing

which includes an evaluation of the computer data processing controls that are designed to ensure correctness and completeness in processing transactions. The other point is the obtaining of evidential matter to afford a reasonable basis for an opinion on the financial statements (the substantive tests)

These two areas are obviously related. The auditor's evaluation of the processing system design and the controls associated with it is a part of his overall evaluation of the controls for each application having audit significance, an evaluation used in deciding on the extent of substantive testing.

### **THE STUDY AND EVALUATION OF INTERNAL ACCOUNTING CONTROL IN EDP SYSTEMS**

The review of internal accounting control in EDP can be divided into a preliminary phase and a completion phase. If the auditor decides to rely on the controls in the EDP system and performs the completion phase of the review, compliance tests are made to test whether the controls were effective for the period under review. This subject is discussed in more detail in chapter 3.

### **SUBSTANTIVE TESTING**

Substantive testing includes analytical review procedures and detailed tests of transactions and balances. The following are some examples of detailed tests appropriate to computer applications:

- Tests for correctness and completeness of data items in records
- Tests to compare recorded balances with physical counts
- Confirmation of transaction amounts and balances with sources outside the organization

These tests can be made by both manual and computer-based techniques.

## **Scheduling Computer-Related Audit Activities**

The timing of computer-related audit activities can follow the typical audit schedule of interim work conducted prior to the end of the financial statement period, period end activities, and post period audit procedures. Period end and post period computer audit procedures require careful planning. For example, a post period procedure to test the accounts receivable file at the end of a period requires a copy of the accounts receivable file for that period. The organization will probably keep a copy of the file for a reasonable time, but the auditor can only be sure of having the data by arranging in advance to have the file retained. The copy of the accounts receivable file (on magnetic tape or disk) is useless without a means for performing analysis. This may be done by printing the file for manual review, by analyzing the file using generalized audit software, or by analyzing the file using computer programs written for this purpose.

The example of an accounts receivable file illustrates areas in which advance arrangements are vital.



Area for Advance PreparationComments

Identification of data processing organization and general procedures (this may have been obtained in the preliminary phase of the review)

The following items may be documented

- Computer hardware and system software being used
- Program library procedures
- Documentation procedures and availability and status of documentation
- File library and file retention procedures

Obtaining and testing of computer programs for audit use

Programs should be obtained and tested well before period end to make sure that the programs can be run and the data requirements can be met

- Computer audit programs written by the client or a CPA firm should be compiled, tested, and maintained by the auditor. The auditor may consider keeping the audit programs under client library control if library controls are adequate
- Compatibility of generalized computer audit software with the client's computer system should be tested. Testing may occur as part of interim procedures, while computer programs for planned period end and post period procedures can be written and tested in advance
- If inquiry software (available with client data management software) is used, the software should be tested prior to audit use

Obtaining of data files for testing

The availability of data files for testing depends on the client's retention procedures. Advance arrangements with the data processing department to retain copies of files or parts of the database should be made early in the planning period. The auditor may arrange to have access to the following

- File layouts and other data documentation for copies of master files and transaction files being used

(continued)

## 24 Planning and Applying Audit Procedures for Computer Data Processing

*Area for Advance Preparation (cont )*

*Comments (cont )*

- Data dictionaries and external schemas describing databases to be accessed
- Copies of master files as of a specific date
- Copies of transaction files for periods being tested plus copies of master files as of beginning and end of test periods

Access to computer facilities on which to execute computer program audit procedures

The client's computer is often used if time is available and there are no compatibility problems with audit software. Alternatively, data may be copied from the client's files or database and taken to a service center or other computer for processing. In either case, the auditor must schedule time for testing programs and for running the program with the data files.

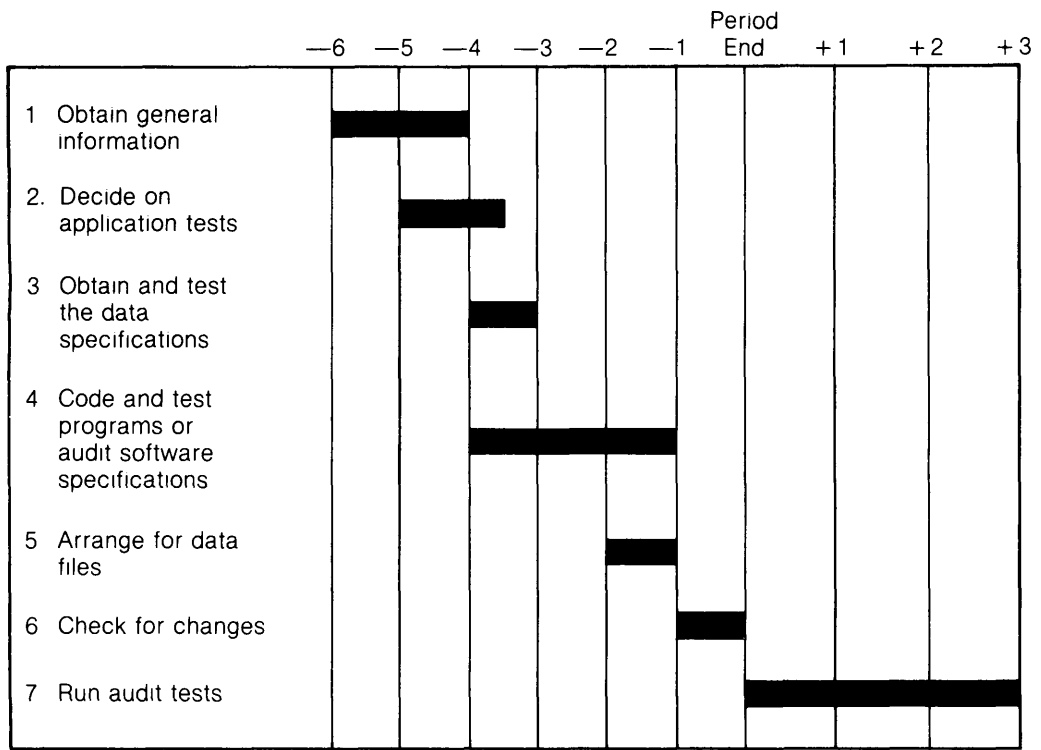
The audit plan should allot time for adequate preparation and testing. The amount of preparation depends on, among other things, the auditor's experience, the complexity of the audit task, and the type of software used. For example, an auditor unfamiliar with generalized audit software should plan for more than one debugging run and should allow for delays in running the job. A unique program written in a language such as COBOL will require time to design, code, debug, test, and document. Depending on the complexity of the program, it will take a programmer approximately one hour to develop and test five to ten program statements in the final program. Based on this rough guide, a small, fairly simple audit program of 150 COBOL statements will require fifteen or more hours for preparation.

The use of audit software requires advance scheduling of audit activities to allow adequate time for planning, specification preparation, program testing, data capture, and running of the software. The chart in table 2-1 illustrates an approximation of the scheduling requirements when the auditor uses computer programs or audit software. The time periods in the figure are unspecified, since these will vary depending on the stability of the client's system, the experience of audit personnel, and so on.

### Summary

The audit activities associated with computer data processing are a part of the total activities in the financial audit. However, because of the characteristics of computer data processing, it is useful to study as a separate topic the audit procedures for EDP systems. There are two major classifications of controls that are useful for this purpose: (1) accounting control and administrative control and (2) general controls and application controls. The study and the evaluation of accounting control establishes the scope of audits of financial statements, therefore, EDP controls that fit within the accounting control classifications are within the required audit scope. General controls and appli-

**Periods Before and After Client Year End**



**Table 2-1** Rough sequence for computer audit activities using computer audit programs or audit software and computer-stored client data

cation controls are the two categories used in the study and evaluation of internal control, general controls provide the environment for data processing, while application controls are associated with a specific application

Within the framework of these major classifications, an auditor may find it useful to categorize a control according to two other classifications (1) development, processing, and continuation controls and (2) preventive, detective, and corrective controls

An assessment of risk of material errors and irregularities may be made at various levels of detail the EDP function, the application, transactions within applications, and the data element The analysis is useful in selecting areas of audit emphasis and in planning audit procedures

The primary aspects of the audit affected by EDP are audit planning, the study and evaluation of internal accounting control, and substantive testing The study and evaluation of internal accounting control have two phases of review a preliminary phase and a completion phase Following completion of the review, compliance testing is used to evaluate whether the auditor can rely on the controls Compliance and substantive tests may be performed manually or by computer

The scheduling of computer audit procedures requires advance preparation to obtain data files and to schedule the time required to prepare, test, and use audit software



---

# The Study and Evaluation of Internal Accounting Control in Computer Data Processing

# 3

The importance of internal control in the audit is defined in the second standard of field work

There is to be a proper study and evaluation of the existing internal control as a basis for reliance thereon and the determination of the resultant extent of the tests to which the auditing procedures are to be restricted [SAS 1, section 150 02]

This chapter describes the application of this standard to the study and evaluation of internal accounting control for computer data processing. The following three AICPA publications are the primary resources for the chapter:

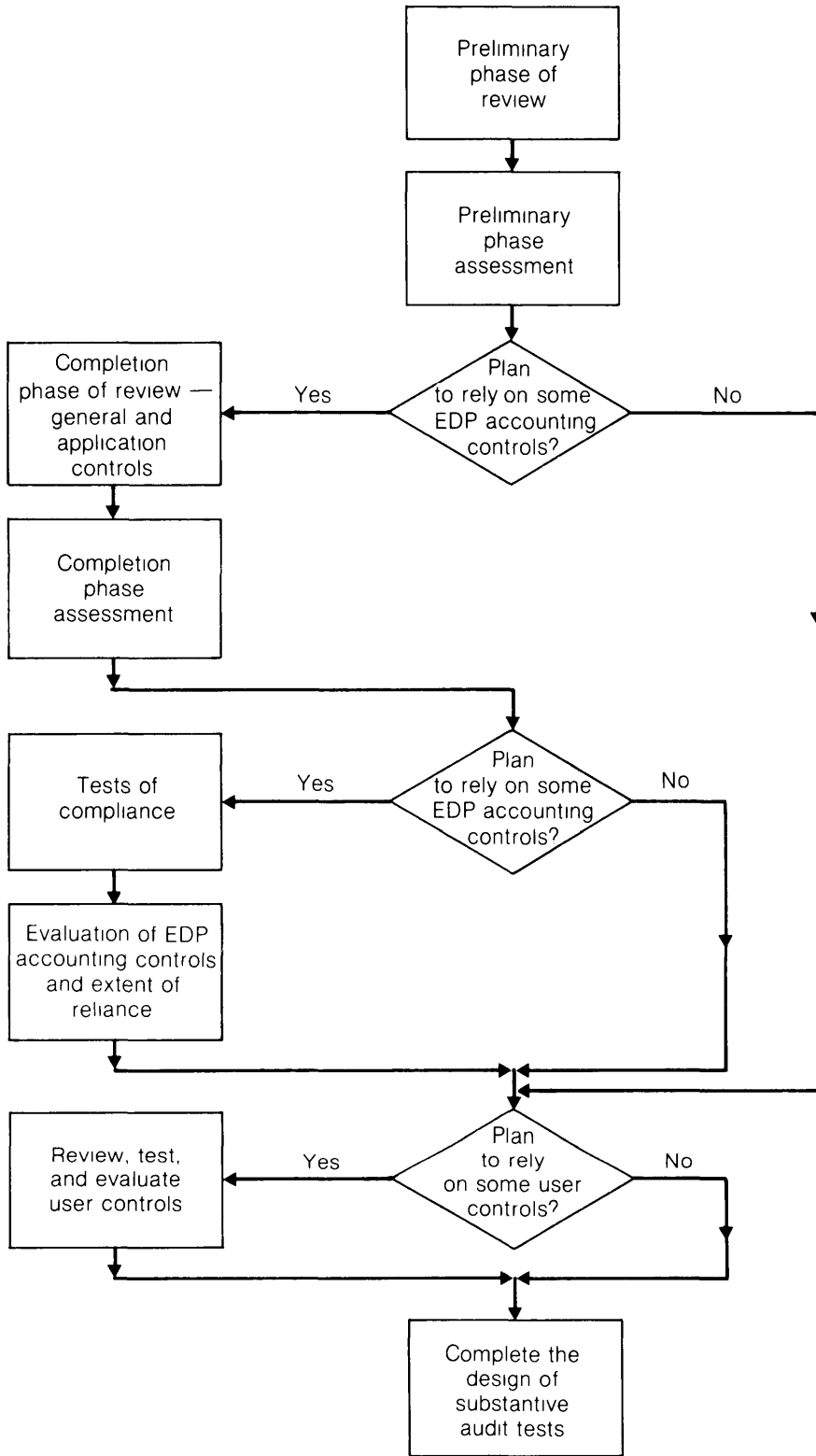
- SAS 1, *Codification of Auditing Standards and Procedures* (1973)
- SAS 3, *The Effects of EDP on the Auditor's Study and Evaluation of Internal Control* (1974)
- *The Auditor's Study and Evaluation of Internal Control in EDP Systems, Audit and Accounting Guide* (1977)

## Steps in the Process

The study and evaluation of internal accounting control can be divided conceptually into four steps, however, for reasons of efficiency and effectiveness, these steps may not remain operationally distinct

- 1 *Preliminary phase of the review* This involves a preliminary review and assessment of the general and application controls in the context of the data processing system (manual and computer)
- 2 *Completion phase of the review* This is a more detailed look at relevant general controls and the application controls for those audit-significant applications
- 3 *Tests of controls* Compliance tests provide assurance that the EDP accounting control procedures have been applied as prescribed throughout the period
- 4 *Evaluation of the system* The auditor evaluates the results of the review and tests of compliance to decide on the extent of substantive tests

The decision paths are diagrammed in figure 3-1



**Figure 3-1** Flow of logic and activities in the auditor's study and evaluation of internal control in EDP systems

## **PRELIMINARY PHASE OF THE REVIEW**

The preliminary phase of the review of the client's system of accounting control provides the auditor with information about the computer data processing system so he can decide how to proceed with the audit. SAS 3, paragraph 25, states, "The preliminary phase of an auditor's review should be designed to provide an understanding of the flow of transactions through the accounting system, the extent to which EDP is used in each significant accounting application, and the basic structure of accounting control." Thus, the preliminary phase has three products:

1. General understanding of the data processing system for accounting records for both EDP and non-EDP activities
2. General understanding of the extent to which EDP is used in significant accounting applications
3. Outline of the basic structure of accounting control for EDP as reflected in general controls and application controls

### **General Understanding of the Data Processing System for Accounting Records**

To determine the scope of audit procedures related to computer processing, the auditor must consider the scope for the review of general controls and the applications to be reviewed and tested. The determination of scope requires a general knowledge of the computer data processing system and the applications for which the computer is used. This background information is obtained by the auditor as part of the survey of a new client's organization and organizational functions or the review of changes in an existing client's organization and functions.

The auditor obtains a general understanding of the data processing system for accounting records by identifying each significant class of financial transactions and reviewing the flow of these transactions as they are processed by EDP and non-EDP procedures. The auditor may gather information by the following methods:

- Discussion with appropriate client personnel
- Review of such documentation as job descriptions, organizational charts, system flowcharts, and systems and procedures manuals
- Observation
- System walk-through (tracing a sample of transactions through the processing system, either by using inquiries or by following such documentation as system flowcharts)

The resulting general understanding permits the auditor to decide which applications have audit significance, since an audit normally does not extend to all applications.

### **General Understanding of the Extent to Which EDP Is Used in Significant Accounting Applications**

In order to understand the extent to which computer processing is used in each significant accounting application, the auditor must also understand the flow of significant classes of transactions through processing activities. This information can be obtained by examining (a) the processing performed by EDP personnel and by computer pro-

### 30 The Study and Evaluation of Internal Control in Computer Data Processing

grams and (b) the division of transaction processing flow between EDP and non-EDP activities. For each application, the study may include the following:

- Type and extent of documentation available for review
- Source document preparation or data capture procedures
- Data preparation or data entry procedures (including access procedures)
- Files used or created
- Various output and reports produced
- Error correction procedures
- Non-EDP processing by recipients of output

#### **Outline of the Basic Structure of Accounting Control for EDP**

The investigation of the flow of transactions and the extent to which EDP is used in significant accounting applications provides the information about the basic structure of accounting control for EDP, which is complemented by a preliminary investigation of both general controls and application controls. General controls are reviewed to provide an understanding of the client's computer data processing organization and the organizational assignments for three major areas of control: development, operations, and data and program files. The significant auditing applications are examined to gain a general understanding of the basic structure of the controls for each of the applications. The application controls can be studied in terms of the flow of processing:

- Control over data recording and data preparation (or direct data entry) to ensure complete and correct preparation of authorized data
- Input controls and input validation procedures to ensure complete input of valid data
- Processing controls to ensure complete processing of data using authorized logic
- Output controls to ensure accuracy and completeness of output and distribution to authorized users
- Error handling for all of the above steps
- Audit trail

#### **Preliminary Assessment**

The review of the processing flow and the extent to which controls are applied are both general at this phase of the review. Documentation of the preliminary phase of the review may include completed questionnaires, simple narratives, simple flow diagrams, or notes identifying and briefly describing the controls in use. These documents document the auditor's understanding of the data processing system, the extent of EDP in each application having audit significance, and the preliminary understanding of general accounting controls and application controls. Documentation should include the planned effect of this assessment on the remaining portions of the audit.

The preliminary phase of the review provides sufficient information on the flow of processing and the structure of controls for the auditor to be able to make a preliminary judgment regarding further audit procedures. The auditor can decide to proceed with completion of the review. The decision may include reliance upon some but not all controls. On the other hand, the auditor may decide not to proceed with completion of the review because internal accounting controls for EDP are inadequate or because he



knows that reliance will not be placed on internal accounting controls for EDP since (a) alternative procedures for obtaining evidential matter are deemed more efficient or more effective or (b) other internal accounting control procedures outside EDP are in existence and deemed adequate

## COMPLETION OF THE REVIEW

The completion of the review supplies the auditor with a detailed understanding of the design of the general controls and the controls in the applications having audit significance. The documentation of the completion of the review is more extensive than the documentation of the preliminary phase, it requires more detailed questionnaires and interviews, as well as more complete narratives, diagrams, and notes. In addition, it usually involves a system walk-through and the observation of the manual procedures and documents used in both the general and application control procedures.

Completion of the review of general controls includes the following

- Organizational and operational controls
- Application system development and maintenance controls
- Hardware and system software controls
- Access controls
- Procedural controls

Completion of the review of application controls includes additional in-depth investigation of the input, processing, and output procedures and controls in each application. The processing steps, audit trail, and control procedures should be documented for each application having audit significance and selected for review. The auditor reviews those controls on which he plans to rely.

After studying the results of the completion phase of the review, the auditor can decide to rely on internal accounting control in computer data processing and proceed to compliance testing, based on the evaluation that controls as designed are adequate. The decision can include reliance on some but not all controls. The auditor also has the option not to rely on internal accounting control in computer data processing, in which case he will not proceed with compliance tests.

## TESTS OF CONTROLS

According to SAS 3, paragraph 27, tests of compliance are designed "to provide reasonable assurance that accounting control procedures are being applied as prescribed. Tests of compliance are concerned primarily with questions (a) were the necessary procedures performed? (b) how were they performed? (c) by whom were they performed?"

The review defines the computer data processing system and the related controls as they are purported to be, that is, as prescribed by organization personnel and EDP documentation. If reliance is to be placed on these controls, the auditor needs to test the functioning of the controls.

Tests of compliance cover both general and application controls. The controls may be implemented by organizational design and visible procedures or by computer program logic. In the case of organizational design and visible procedures, tests of compliance use traditional techniques, such as the following

## 32 The Study and Evaluation of Internal Control in Computer Data Processing

- *Observation* Examples are observation of activities of the control group the librarian, and computer operators
- *Corroborative inquiry* An example is interviewing personnel at the operational level regarding the control activities
- *Inspection of documents containing evidence of compliance* Examples of such documents are error and control listings, program change documentation, library records, and error logs Performance may be demonstrated by the data on the documents or signatures or initials of preparers and approvers

The auditor obtains information on processing logic and programmed control procedures from interviews with personnel and inspection of documentation To verify his understanding of the system, the auditor obtains evidence of program logic and program changes by performing the following

- Procedures to obtain evidence that programs contain processing logic and control procedures described in documentation and by organization personnel These procedures may include—
  - Observation of manual processing and control activities that provide evidence of computer program logic
  - Examination of evidence of computer program logic, such as data on transaction documents (when compared with input data), error reports, and control reports
  - Verification of program logic through techniques for program examination and testing
- Activities to obtain evidence that no unauthorized or undocumented changes have been made in the programs for applications within the scope of the audit These procedures may include—
  - Corroborative interviews with operational personnel regarding program change controls
  - Observation of segregation of functions and handling of program change by programmers and the program librarian
  - Examination of documents that indicate compliance with program change controls, such as program change documents and program library records, and tracing of specific program changes to change documentation
  - If the foregoing audit procedures are not adequate, substantiation of program changes through examination and testing procedures This can include the procedures used for testing of original program logic plus library procedure tests In some cases, an auditor may maintain a controlled program copy and compare this with the library copy

### **EVALUATION OF ACCOUNTING CONTROL IN THE DATA PROCESSING SYSTEM**

The evaluation of accounting control within the EDP system is combined with an evaluation of controls performed by users of data processing and an overall evaluation of accounting control for the data processing application (see figure 3-1) This assessment of the extent to which the combination of EDP system controls and user controls may be relied on provides the basis for determining the substantive tests to be performed

## Applying the Distinction Between the Two Phases of the Review

To distinguish the preliminary phase of the review is to recognize that the auditor has an option regarding the completion of the review. However, the distinction may be difficult to apply. It is sometimes difficult to make a clear operational distinction between the two phases of the review, and it is often efficient to obtain both the general information required for the preliminary phase and detailed information for completion at the same time or from the same person. For example, it may be efficient to use a single questionnaire. The information gathered for the preliminary phase of the review is obtained in the first review for a new client, but in subsequent audits the preliminary phase can be limited to identification of changes made since the last review.

The operational distinction between the preliminary phase and completion of the review is suggested by the type of questions asked, the sources of information, and the audit procedures used.

<u>Bases of Distinction</u>	<u>Preliminary Phase</u>	<u>Completion</u>
Questions asked	Do controls provide ? Are there procedures for that provide control over ? Are there controls over ? Are there controls that ensure ?	What are the controls and how do they provide ? What procedures provide controls over ?  What are the controls for ? What are the controls and how do they ensure ?
Sources of information	Client personnel, organizational charts, and job descriptions, application system documentation, source documents, descriptions of system flow and files used, descriptions of error handling procedures, output documents	Same as for the preliminary phase, plus more detailed system documentation, more detailed procedure descriptions, and additional client personnel
Audit procedures	Interviewing, observation, review of documents, system walk-through	Same as for the preliminary phase, plus more detailed information about performance in order to document "what" and "how "

The difference between the inquiries made for the preliminary phase and those made for completion is illustrated by the following example—the review of general

### 34 The Study and Evaluation of Internal Control in Computer Data Processing

controls for application program maintenance (corrections and enhancements) (See also Appendix A questions in section 2 2 3 )

#### Questions at the Preliminary Phase

- Do you have formal procedures for control over application program maintenance?
- If yes, describe the overall structure of the controls and the general flow of the control activities

#### Questions at Completion of the Review

- Describe how application program maintenance is organized
- Describe the specific procedures followed and show the forms used for—
  - Program maintenance requests
  - Approval procedure to authorize work
  - Documentation of changes made and testing performed
  - Review and approval of changes, testing of changes, and documentation of changes by the data processing reviewer and by the user who requested the change
- Describe the specific control procedures for ensuring that—
  - Updated programs that supersede prior versions are placed in operation
  - Documentation of changes is performed adequately and completely

The distinction between the preliminary phase and completion of the review does not recognize differences in the procedures necessary for a first audit and procedures in continuing audits. These differences can be significant.

*First audit* Understand the accounting system for both EDP and non-EDP segments in the following areas

- Flow of transactions
- Extent to which EDP is used in significant accounting applications
- Basic structure of accounting control, EDP controls, and user controls

*Continuing audit* Identify changes

### The Optional Review of Provisions for EDP System Continuation

The review and testing of provisions for system continuation, if included in the scope of an audit engagement or conducted as a special study, generally include consideration of physical disaster protection, control over access to facilities, programs, and data, and backup and recovery provisions. Lack of provisions for continuation would not ordinarily affect the presentation of audited financial statements, however, contingency planning should be considered when there is more than a remote chance of physical disaster. (See FASB Statement 5.)

The review may be divided into (1) general continuation provisions and (2) continuation controls and procedures specific to an application. There are general provisions for physical disaster protection and recovery, and backup and recovery. The continuation procedures specific to an application usually include special access controls and backup and recovery procedures. Provisions such as breakpoints, from which inter-

rupted processing may be restarted and provisions made for efficient recovery in the event of processing failure or file destruction, are incorporated in the individual applications

The review of physical disaster protection covers precautions against fire, flood or other water damage, wind damage, earthquake (if relevant), and riot. The protection provisions include devices and procedures to prevent, detect, and contain the destructive event and to minimize damage to the organization's data processing capabilities.

Review of backup covers retention and storage of backup material, including up-to-date copies of all programs, data files, procedure manuals, system software, and input and output forms, which should be stored in a secure location, preferably off-premises. In regard to recovery provisions, the auditor reviews operational recovery procedures, as well as the disaster plan and the extent to which it has been tested under simulated disaster conditions.

The review of continuation provisions can be a separate study, or it can be included in the inquiries and questionnaires in both the preliminary phase and completion of the review of controls. Tests of compliance for continuation provisions usually consist of observation of the existence and readiness of physical disaster protection and access control devices and procedures. Backup and recovery provisions can be tested by observing records and by asking for specific items needed in a recovery procedure.

## Summary

The auditor's study and evaluation of internal accounting control in EDP can be divided into four parts:

- Preliminary phase of the review
- Completion of the review
- Tests of compliance
- Evaluation of the system

The preliminary phase of the review aims at a general understanding of the flow of transactions through the accounting system in both EDP and non-EDP processing. The extent of EDP use is established for each significant accounting application. A general understanding is also obtained of the basic structure of accounting control for computer data processing, the general controls, and application controls for each significant accounting application. An assessment at the end of the preliminary phase results in a decision to proceed with completion of the review and evaluation of internal accounting control because of expected reliance on internal accounting control or a decision not to rely on EDP controls. If a decision is made to rely on the internal accounting control in EDP, completion of the review requires identification of the specific characteristics of the general and application controls. If reliance is still planned, the controls are tested for compliance. Compliance tests are followed by an evaluation of internal accounting control in the EDP system and the design of substantive tests.

The two conceptual phases of the review may not always be distinguishable in practice because it is often efficient to obtain both preliminary and detailed information at the same time in a single interview or in a single inspection of documentation.

The study and evaluation of internal accounting control may be extended to include a review and testing of provisions for system continuation. The continuation controls are either general (being applicable to a number of applications) or specific to an application. The review covers physical disaster protection and backup and recovery provisions.



---

## SECTION II

# General Controls for Computer Data Processing

In section II, general controls, which provide a controlled environment in which applications can be processed, are explained

The general controls have been classified by the audit and accounting guide, *The Auditor's Study and Evaluation of Internal Controls in EDP Systems*, into five categories, the chapters in this section in which the controls are discussed are not organized strictly around the five categories of general controls but according to logical groupings for presentation

<u>Category of General Controls</u>	<u>Chapter</u>
1 Organization and operation controls	Chapter 4—Preferred Practices in Organization and Management of the EDP Function
2 Application system development and maintenance controls (including documentation)	Chapter 4 and Chapter 5—Documentation of the Data Processing System
3 Hardware and system software controls	Chapter 6—Hardware for Control Over Equipment Malfunctions and Chapter 7—System Software and Database Management System Features for Control
4 Access controls	Chapter 8—Safeguarding the Availability, Access, and Use of Computer Facilities, Programs, and Data
5 Procedural controls	Chapters 4 and 8 for general procedural controls (Procedures relating to input, processing, and output for applications are explained in section III )





---

# Organization and Management of the EDP Function

# 4

An organization's data processing installation should be organized and managed by the same methods that have proved effective in the enterprise's general business practice. There should be a plan of organization, clear assignment of responsibilities, and segregation of functions. A master plan should govern the acquisition or retirement of hardware, software, and application systems. Application system development projects should be carefully managed and controlled, and there should be written policies and procedures for development and implementation of applications, which, in turn, should be used to measure and evaluate performance.

## Plan of Organization

In data processing, as in other business operations, it is necessary to establish an organizational structure that defines individual responsibilities for all functions. The design of the organization and the assignment of responsibilities should provide for division of functions for purposes of both internal control and efficiency.

### JOB DESCRIPTIONS

To establish responsibility, job descriptions should be prepared for all data processing personnel, stating the job titles and describing their functions. Employees should receive a copy of their appropriate job descriptions and should receive instruction in all factors related to their work assignments. Although titles may vary among installations, the following brief job descriptions cover the most common data processing positions.

*Systems analyst.* Analyzes the requirements for information. Evaluates the existing application system and designs new or improved data processing procedures. Outlines the new application system and prepares specifications that guide the programmer. Develops implementation plan and procedure manuals.

*Systems programmer.* Supplies technical guidance concerning the operating system to all other members of the data processing staff. Provides a link between the needs of application systems and the requirements of the operating system. Modifies, tests, and maintains all nonapplication software supplied by software vendors.

*Applications programmer.* Determines the logic of the computer programs required by the overall system designed by the systems analyst. Codes the logic in the computer program language. Debugs the resulting program and prepares documentation.

## 40 Organization and Management of the EDP Function

*Computer operator* Operates the computer according to the operating procedures for the installation and the detailed, written procedures for each program

*Librarian* Maintains the library of all data and program files Releases the files and follows up on their return in accordance with established policy Often is responsible for maintaining a library of all documentation

*Control clerk* Controls, and sometimes schedules, all data entering the data processing system Reviews output, performs control balancing, and distributes reports coming from the system Maintains error logs

*Database administrator* Controls all aspects of the design and development of the database Provides continuing liaison between the EDP design teams and the users in regard to logical file structures and data content Responsible for the preparation of the data dictionary

*Data entry operator* Prepares data for machine processing by entering it via a keyboard into a device that will either record the data on machine-readable media (cards, tape, or disk) or enter it directly into the computer for processing

As demonstrated in the sample organizational charts, the functions of these positions may be combined or separated In general, the position of database administrator requires the broadest background and highest qualifications The systems analysts and programmers require the next highest level of qualification, while the computer operator requires less training and experience than the other positions Many organizations make a distinction between analysts and programmers, and others have positions called "analyst/programmer" or "programmer/analyst "

### SEGREGATION OF FUNCTIONS

When a company establishes a data processing organization, proper recognition should be given to achievement of internal accounting control, important to which is segregation of functions The same principle that calls for a separation of the functions of record-keeping, operations, asset custody, and internal auditing also applies to the separation of basic data processing functions

The three major data processing functions are (1) application system development and maintenance, (2) operations, and (3) control Six data processing functions are listed below within those groupings The first five functions are common to all computer data processing installations, the sixth applies to an installation with a database management system

<i>Data Processing Function</i>	<i>Position</i>
1 Application system development and maintenance	
● Systems analysis and design (1)	Systems analyst
● Programming (2)	Programmer
2 Operations—Machine operations (3)	Computer operator
3 Control	
● Processing control (4)	Control clerk or control group
● Custody and control over files, media, programs, and documentation (5)	Librarian
● Management and control of the databases (6)	Database administrator

It is a good idea to separate the three major functions, and internal accounting control is strengthened if all six functions are separate and distinct. This separation may also result in operational efficiencies because the functions require different levels of training and skill. It may be satisfactory to combine the development activities of systems analysis and programming under a single supervisor or, in small installations, to include both activities in a single job. It is not desirable to overlap development activities with machine operation, although it may not be possible to avoid this in small installations.

A suitable plan of organization and division of duties is important because computers usually cause concentration of data processing activity in a smaller number of people than would be required for an equivalent manual system. The system is exposed to manipulation and fraud if within it a single person has both operational knowledge and access to procedures and programs.

### **SEGREGATION OF FUNCTIONS BETWEEN EDP AND USERS**

The organizational planning should segregate functions to avoid EDP performing user functions. In the case of EDP personnel, the following duties are considered incompatible for control reasons:

- Transaction origination or correction
- Transaction authorization
- Initial data preparation
- Custody or control over non-EDP assets
- Authorization or change of controls
- Origination of master file changes

In some cases these functions are performed by computer programs. An example is origination of a purchasing transaction based on computer program logic. In such cases, the automatic performance of the function should be reviewed by personnel outside EDP using output from the programmed execution.

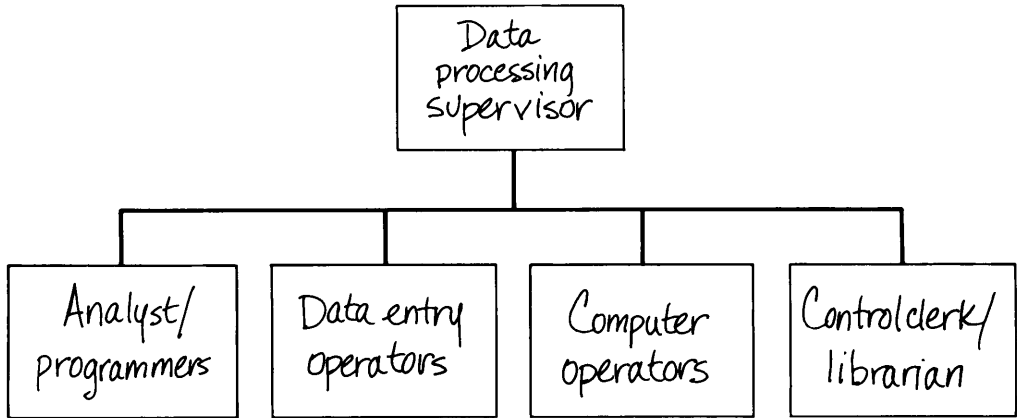
The incompatible functions should be segregated, and control procedures should be established to detect violations of the segregation. An example is user responsibility for master file data and user review of master file changes and of values existing in master files.

### **TYPICAL ORGANIZATIONAL CHARTS**

Three charts presented here illustrate the typical organizations for small, medium, and large computer installations.

These charts do not indicate the location of data processing within the overall organizational plan because this varies from company to company. In most small organizations, data processing is the responsibility of the chief financial or accounting officer, such as the financial vice president, the treasurer, or the controller. In many large organizations, the data processing system has been separated from finance or accounting, and the head of EDP has a title such as "vice president of corporate information systems."

The computer-based information system is organized as a service center to satisfy the data processing needs of the entire organization. By maintaining data used by many departments, data processing becomes involved in activities that cross existing divisional or departmental lines of authority. Therefore, data processing personnel



**Figure 4-1** Organization chart for a small data processing installation

should report to an executive who has authority to resolve possible conflicts. This generally means that the executive in charge of data processing should be on an equal level with the heads of the departments being served.

Figure 4-1 shows an organizational chart for a small installation. Note that the systems analysis and programming functions have been combined. One employee is assigned the duties of both control clerk and librarian.

Figure 4-2 illustrates a medium-scale installation. In this installation, the functions of the systems analyst are separated from those of the programmer, and a control group, a librarian, and a systems programmer have been added to the system. In this expanded organization, there may also be opportunities for a hierarchy of supervision. For example, the programming staff might include senior programmers, programmers, and trainees.

Figure 4-3 shows an organizational chart for a fairly large installation. The increased level of specialization in such an organization is readily apparent. Programming is divided into two groups: systems programming and applications programming. The analyst group contains both project leaders and specialists. The specialists (forms design, hardware, and so on) are assigned to projects when they are needed. Some organizations separate new application development from the maintenance of existing applications. In such an organization, database administration is a separate function.

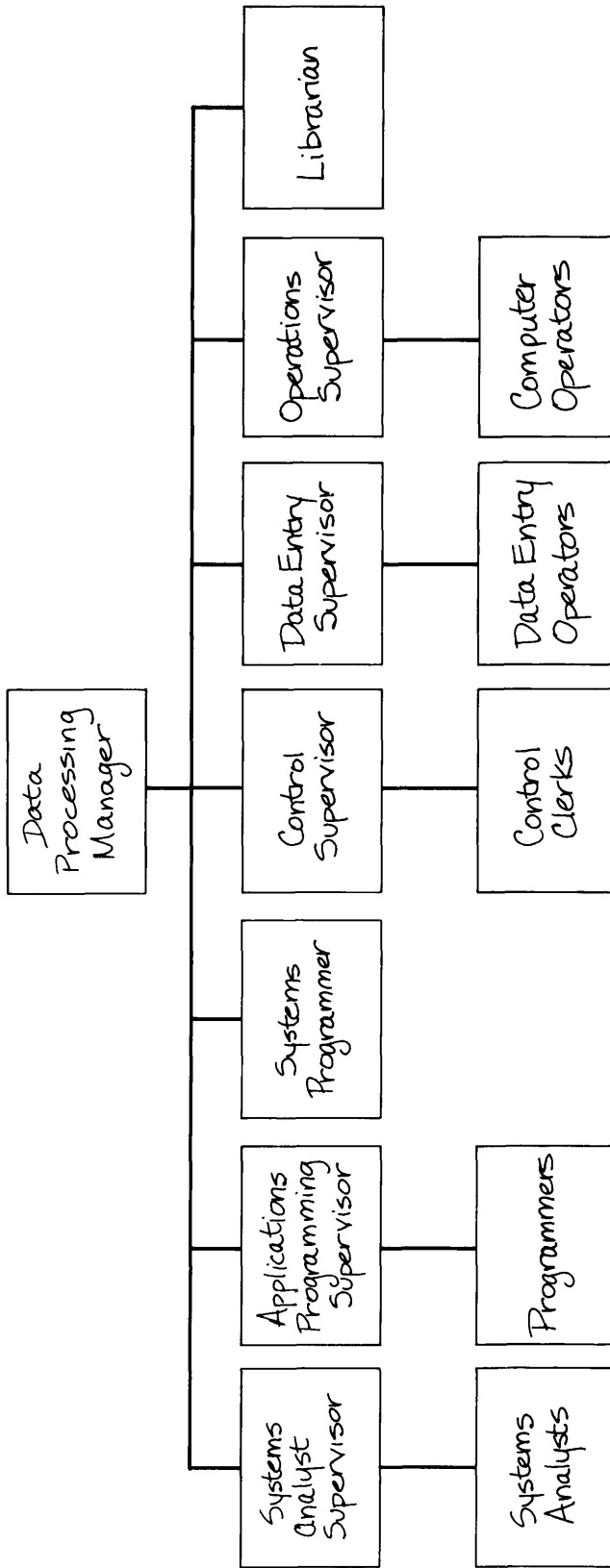
Four functions shown in figure 4-3 may be found in large data processing operations, but they are not yet commonplace.

*Quality manager* Is responsible for establishing and enforcing policies governing systems design, programming, and documentation. Reviews and approves final documentation. Evaluates compliance with policies and procedures. May be responsible for reviewing the adequacy of system testing.

*Security officer* Is responsible for establishing policies and procedures related to physical and data security. Investigates all security violations and, as required, recommends corrective and disciplinary action. Controls the issuance of passwords, badges, keys, and other access control devices.

*Network administrator* Is responsible for the design, selection, installation, and management of data communications facilities. Maintains liaison with communications vendors.

*Schedulers* Are responsible for scheduling production runs and special requests.



**Figure 4-2** Organization chart for a medium-sized data processing installation

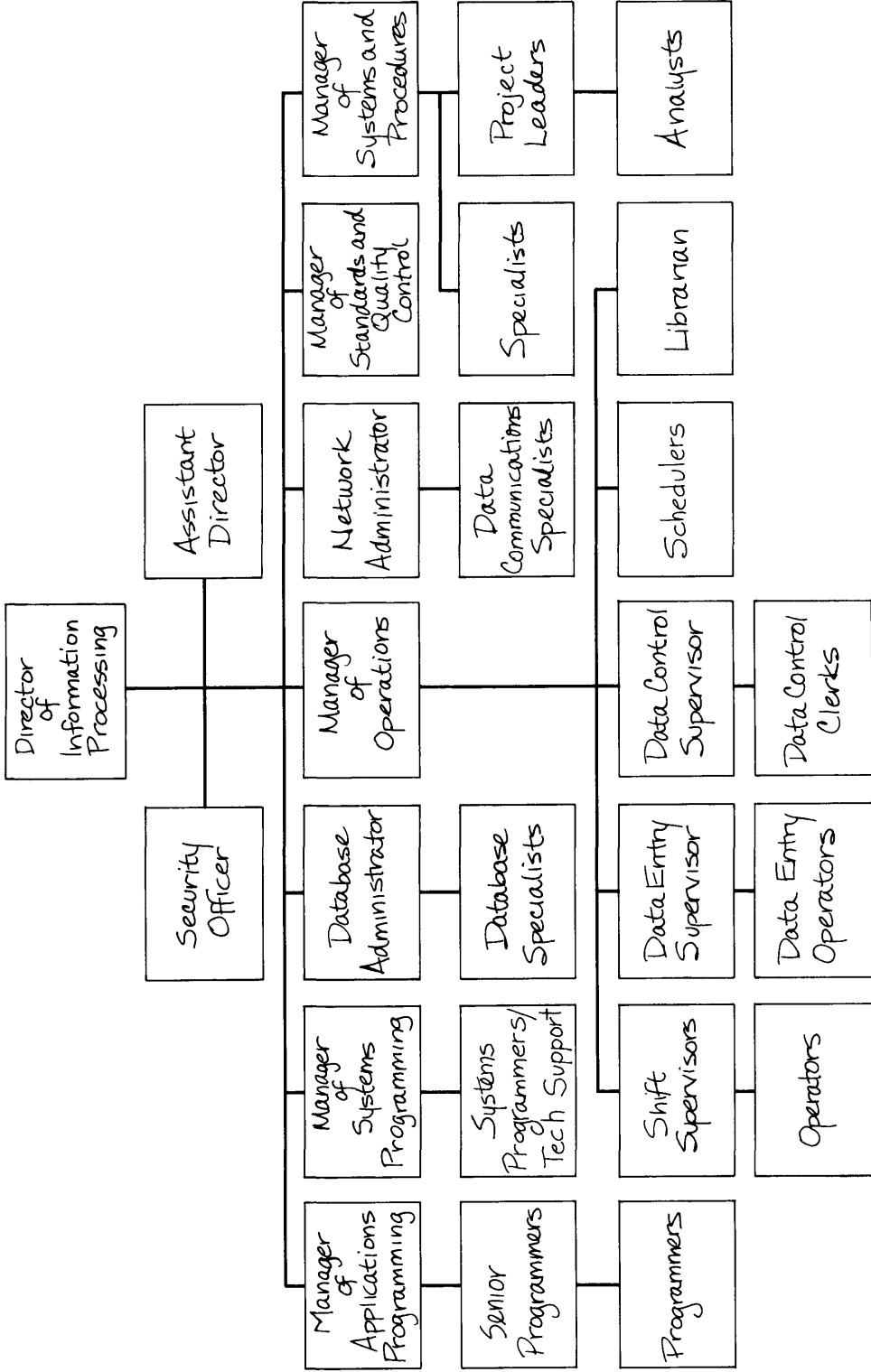


Figure 4-3 Organization chart for a large data processing installation

## The EDP Control Function

The plan of organization, division of functions, and operating procedures should provide for an EDP control function. The EDP control function is subdivided into control activities internal to data processing and control activities outside data processing. The major areas of responsibility in the control activities internal to data processing are processing control, library control, and database administration.

### OUTSIDE DATA PROCESSING CONTROL

Outside control provides an independent check on the functions of the data processing department by a user department or by a separate control group. The user department performs a control function by maintaining independent data to check the results of processing. For example, if the general ledger is maintained through the computer, the accounting department can keep a control total of the debits and credits to be posted by the computer to the general ledger. The updated general ledger from the computer should show a total change equal to the control total recorded by the accounting department.

A separate, independent quality control group may be established within the user area to perform a continuing evaluation of data processing output when there is a large volume of data to be controlled. For example, a large corporation may have a payroll processing control group responsible for evaluating the payroll data produced by the computer. This could be done by performing various tests on the data and by using control totals. The outside control function, as typified by the evaluation group, should be under the direction of accounting, finance, or some other function that is able to provide an independent and critical review of performance.

### PROCESSING CONTROL FUNCTIONS

The control of processing includes receiving data to be processed, monitoring progress of processing, and controlling distribution of output. Typical job titles are "processing control" and "input/output control." The duties include recording and checking all control figures used to ensure completeness and correctness of processing.

The activities of the control clerk or control group should be specified both in the systems and procedures manual and in documentation for each computer application. In general, the control function, regardless of the number of people involved, includes the following duties:

- Logging of input data
- Recording of control information
- Recording of progress of work through the department
- Reconciliation of computer-produced control information with other control information
- Supervision of the distribution of output
- Scrutiny of console logs and printed control information in accordance with control instructions
- Liaison with users regarding errors
- Logging of correction requests and corrections made
- Scrutiny of error listings
- Maintenance of the error log or error report

## 46 Organization and Management of the EDP Function

The work of the processing control group is documented by the maintenance of the logs, the signing or initialing of control documents, and the review of control reports. For instance, if a transaction file is processed with a master file to produce an updated master file, the sum of the transaction file and the master file should equal the total of the updated master file. The person assigned the processing control is responsible for making or reviewing such a comparison.

### **LIBRARIAN CONTROL FUNCTION**

The librarian control function maintains control over data and program files. The control includes custody and control of the file storage media (disks and tapes).

In a large installation, a full-time librarian may control access to data and program files on each shift. As an alternative, a librarian on the first shift may set up all the files to be used on the other shifts and entrust them to an employee who will exercise physical control. If a file that was not provided is needed, only one supervisor (without operations responsibility) on each shift should have the authority to enter the library and obtain the needed file. Because a small installation may not be able to justify a library function on a full-time basis, someone can be assigned to this duty as a collateral responsibility. Typically, a control clerk or the data entry supervisor might be called upon to perform as a part-time librarian.

The librarian may also be assigned the control of documentation. Documentation is important in the development, operation, and maintenance of applications. Further, documentation is often used as a basis for management control and review of application development activities. Once it has been prepared, reviewed, and approved, this documentation should be placed under the custody of the librarian function.

### **DATA ADMINISTRATION FUNCTION**

If databases are used, an individual may be assigned to the position of database administrator (DBA). The experience of several installations that have made a successful transition to a database indicates that an effective DBA is vital. In the plan of the organization, the DBA should report to the highest level within EDP. Ideally, the database administrator should report to the executive with overall responsibility for data processing.

Typical duties of the database administrator may include the following:

- Evaluating and selecting database management software
- Developing definitions of data items
- Designing the physical and logical structures of the database
- Establishing data naming standards and conventions, including the actual assignment of names
- Controlling the conversion of existing files to database formats
- Setting rules for input validation and updating
- Controlling the preparation of all software that will be used to perform database updates
- Implementing access control and other security provisions to ensure the integrity of the database



- Documenting the database, including preparing and maintaining the data dictionary and advising users and analysts about the use and interpretation of data elements
- Providing for checkpoints, processing restarts, and recovery of the database
- Analyzing all revisions of database structure and content to assess their impact on related application programs
- Resolving conflicts among users, data processing operations, and the systems design staff

From this list, it is apparent that the database administrator is an important control function. However, some organizations feel that the database administrator has too many responsibilities. To reduce this concentration, some organizations will not allow the DBA or the DBA staff to have access to applications software. User-oriented reviews of output data and internal audit procedures are also used to maintain a balance of database control.

## Management of Computer Installations

The same principles that govern general business management also apply to computer data processing. As it does in non-EDP areas, management often prepares and uses a systems and procedures manual, which is useful in training and supervising personnel and in evaluating their performance. Because the use of differing conventions in programming, documentation, and operations can create confusion, a manual that describes standard procedures and conventions for the particular installation can be a valuable management aid. The systems and procedures manual normally covers organization and personnel, operating procedures, and standards.

### PERSONNEL

The quality of data processing is directly related to the capabilities of its personnel. Several equipment manufacturers, as well as independent organizations, have developed aptitude tests that can aid in evaluating prospective employees. Some companies ask prospective programming and systems analysts to solve a practice problem or to write a short program to demonstrate their competence.

Personnel practices and methods of evaluation should be codified in the systems and procedures manual. Employees should know what is expected of them and how their performance will be measured.

Personnel should follow a schedule that governs all phases of their work. Programmers and analysts should be assigned to specific jobs, given job budgets, and required to render periodic progress reports. Operators should be assigned to specific runs with specific equipment and should be required to report reasons for schedule deviation. To improve control, operator assignments to specific jobs or applications should be rotated periodically. Of course, these procedures will vary with the size and structure of the organization, but the concept of rotation in scheduling is applicable to even the smallest EDP organization.

## **OPERATING PROCEDURES**

The operating procedures section of the systems and procedures manual describes specifications for machine operation, hardware maintenance, scheduling, file retention, housekeeping, job record-keeping, emergency procedures, and physical security. For machine operation, the manual includes standard operating procedures for all equipment to ensure that uniform techniques are employed. In the area of hardware maintenance, it provides schedules and procedures for the normal cleaning of tape and disk drives, printers, and other equipment. Concerning scheduling, it indicates specific time slots for normal processing, program compilations, debugging, testing, and preventive maintenance. A certain amount of slack time should be provided to handle the unscheduled reruns that are required. For file retention, the manual details file control and retention techniques. Concerning housekeeping, it includes procedures for handling files, using supplies, storing programs, and keeping the data processing center neat to avoid loss or destruction of programs and data. For job record-keeping, it lists procedures for control over jobs or programs received for processing and for the return of completed jobs to the user. The manual should include plans and procedures for emergencies, such as flood, fire, and power failure, including the steps required to perform an emergency shutdown of the system. Plans for providing backup for hardware, software, and data files should be described in detail. A list of the phone numbers of key employees and suppliers should also be included. Finally, the manual should describe the procedures and safeguards designed to provide physical security for the computer installation.

## **STANDARDS**

This section of a computer installation systems and procedures manual standardizes vocabulary, programming, debugging, and documentation.

It might include a description of the forms, symbols, and conventions used in flowcharting. Generally, the conventions defined by the American National Standards Institute should be employed. (See Appendix B.)

A section on program coding conventions covers the standardized data and procedure names used by the installation, standardized abbreviations, and such conventions as the method to use in differentiating letters from numbers. It may also cover how the program statements should be formatted and how statements are to be numbered. Restrictions or prohibitions on certain coding options may also appear, this often involves a ban on the use of particular verbs or structures that are known to be inefficient or troublesome.

The manual might cover standard glossary and standard abbreviations. Abbreviations are especially valuable for flowcharting. A list of standard abbreviations should be prepared, and the use of nonstandard alternatives should be prohibited. Any word that does not have a standard abbreviation should be spelled out. The glossary should specify the meaning of special terms commonly used in the installation. If there is confusion in the industry regarding the exact meaning of a term, the definition used by the installation should be given in the glossary.

Programming languages, such as COBOL, permit the use of reasonably descriptive variable names (to a maximum of thirty characters). This feature enables a programmer to write programs that use descriptive names to identify the data being processed, which makes it much easier to follow the logic of a program.

Some programmers have a tendency to use short names in order to reduce coding time, and software packages are available that convert shorthand into the correspond-

ing full name, which will be used in all program listings and documentation. For example, the programmer might write CNA-1, but the program would reference CUSTOMER-NAME-ADDRESS-FILE-01.

Within an installation, it is normal to develop standard techniques for performing some common computer operations used by many programs, and this is another topic that may be covered in the systems and procedures manual. For example, routines to clear storage, to print standard headings, or to calculate a storage address on a direct access file may be used in hundreds of different programs. These routines can be stored in a machine-readable format and incorporated into other programs as they are required, obviating redesign or rewrite of the set of instructions every time it is needed for a new program. These modules should be documented and subject to rigid rules to prevent their misuse.

Some installations prohibit the use of certain types of coding or particular COBOL verbs. It is often difficult to get programmers to observe such standards. Further, it is difficult to review the code for such violations because of the large volume of programs that must be checked. However, packages can perform this review quickly and efficiently. The user can specify coding features that are forbidden and can, in some cases, specify the alternate coding that should be used. When a program enters the system, it is processed through this edit package. If a restricted verb is picked up, it will be replaced by the alternate coding. If no alternative exists for the prohibited code, it will be flagged and reported. As long as a program contains one or more error flags, the edit software will not allow that program to be compiled.

The manual should detail standard job control language procedures. Most computer systems are run under the control of an operating system, the interface between application programs and the operating system is accomplished by the use of some form of job control language. The job control language is used to instruct the operating system in processing a particular job or program. The preparation of the proper control statements can be a complex undertaking and often involves a depth of technical knowledge beyond that of many programmers, as a result, it is common practice to have standard sets of control statements written to perform common system functions and to have these statements stored on a machine-readable library. When a programmer wants to perform one of these standard functions, the proper set of statements within the library can be called up. A clear description of each set of statements and its use should be included in the manual.

Control language generators can be used to prepare the initial setup of the required operating system instructions. These generators simplify the task of preparing control language statements and can be used to help enforce standards that govern the coding of such statements in an installation.

The manual should list the procedures to be followed in debugging, which include specifications for desk checking, program compilation, and approval process to be followed before a program is released. Desk checking (the process of reviewing the documentation and manually tracing the path of data through the program to see if the logic is sound) is a good practice before program compilation is attempted. The approval process should be specified in detail and should include a provision for obtaining the user's approval of testing for all programs and subsequent modifications.

Documentation standards should be specified in detail. (See chapter 5.)

Finally, the manual should describe the procedures to be followed and the review and test functions to be performed each time a program is changed. For instance, a change log should be maintained for each program, and an entry should be made to record the nature, effective date, and supervisory approval of the change.

## Master Plan

A long-range master plan should be prepared to provide a framework for EDP development. Because of the rapid pace of technological developments, it is difficult to make definitive plans. Therefore, the planning effort in data processing must be flexible but should be similar to that employed in any well organized business. The master plan should be based on the plan of the organization and be approved by top management.

The EDP plans should cover a two- to five-year period. In some cases, a longer period may be justified. For example, an organization with a seven-year lease on its hardware mainframe might develop a two- to seven-year plan. Long-range plans in the EDP areas should be reviewed and revised each year.

The master plan for EDP should cover five basic areas:

1. **Application systems.** There should be an inventory and a review of existing application systems, a projection of future modifications of existing systems, and descriptions of new systems that need to be developed. Starting and completion dates, cost projections, file sizes, and estimates of processing time should be part of this section of the plan.
2. **Software.** Future requirements for operating systems, compilers and assemblers, and other systems support or utility software should be estimated. These plans will be directly related to those developed for application systems. While many of the factors that influence software plans are beyond the control of the organization doing the planning (for example, a vendor's release date for a new or improved operating system), it is still useful to formalize future intentions, even if only tentatively.
3. **Equipment.** In light of application and software plans, long-range planning should cover projected hardware upgrades, additions, and replacements. Plans for implementation and conversion should also be prepared.
4. **Personnel.** Requirements for staff changes, additions, and training should be outlined. Training plans should be carefully coordinated with application development and plans for software and equipment acquisition.
5. **Budget.** The financial requirements in order to carry out the plan are summarized.

## Management and Control of Application Systems Development

Although operating failures usually create more headlines ("City Loses \$7 Million in Computer Goof"), the failure of an applications systems development project can be just as costly. Many firms have invested substantial amounts of money in data processing projects that were never implemented. Since the cost of failure in this area can be quite high, management should develop control over the application systems development process to ensure timely, orderly, documented development of applications that are responsive to user needs and organizational objectives for cost, control, auditability, and maintainability. Such controls generally employ the concept of a system development life cycle plus methods for quality assurance in programming. Generally,

these management and control processes require user involvement for them to be effective, and auditor involvement further enhances their effectiveness

### **APPLICATION DEVELOPMENT LIFE CYCLE**

The development life cycle consists of a sequence of phases. Each phase has objectives, processes, products, and reviews. The reviews provide a mechanism for determining at each phase whether user needs are being met and whether objectives regarding cost, control, auditability, and maintainability are being achieved. There are several different labels and descriptions for the application system development life cycle, but they all follow a pattern similar to the one presented here, as table 4-1

### **QUALITY ASSURANCE IN PROGRAMMING**

Several methods are used in organizing and managing computer programming to achieve results that are error free and within time and cost estimates. Management may assign each programming project to a team led by a chief programmer, who defines modules, codes main control routines, and reviews code written by team members, the team also includes a librarian, who handles all test runs and documentation. Alternatively, management may arrange for a formal, constructive peer review of program design, program code, and documentation, called a "structured walk-through." It may set up a development support library to maintain records, documents, coded program modules, test data produced by the programming team, and software to maintain program records

An important method for achieving quality is modular or structured programming. In this approach programs are designed as organized collections of small program units ("modules"), which are called into operation in the proper sequence to perform specific tasks. The job of sequencing these calls is performed by a master module that is often called the "nucleus" or "main line module." Although each module may perform one or more functions, the most effective program is one in which one module performs one function, and there are limits to the maximum size of these modules. For example, COBOL modules are often limited to one page of code (forty to fifty lines). A module accepts only specified input supplied by the nucleus or by another module, and it will return output and control to the calling nucleus or calling module.

This approach encourages the development of standard modules that can be used by all programs that require the same function, for example, testing a field for the presence of all numeric data, writing a line on the printer, or reading a master file. Use of standard modules can reduce the time needed to complete a programming assignment, since, as the library of building blocks increases, more and more of each new program is available "off the shelf" and will not have to be created anew.

The modular approach also simplifies testing and improves reliability. Since each module is designed to be an entity in itself, it can be tested as a separate unit. In this way, it is possible to adopt a piecemeal approach to testing. When a module has been tested and placed in the standard library, the need to test that module when it is used in a new program is reduced. Since modules are easier to test than completed programs, they usually produce a program that is more reliable than one created from scratch. Further, since modules are each designed to perform a specific function, it is often easier to locate a problem when an error is detected. As a result, modular programs are easier to maintain.

Phase of Application Development Life Cycle	Product	Description
Feasibility assessment	Feasibility report	An examination of the project in regard to four types of feasibility: <ul style="list-style-type: none"> <li>● <i>Economic</i> Does the monetary benefit exceed the cost?</li> <li>● <i>Technical</i> Is the project within the existing state of the art?</li> <li>● <i>Operational</i> Can the application perform the required function within the organization?</li> <li>● <i>Schedule</i> Can the project be completed in the time required?</li> </ul>
Information requirements determination	Information requirements report	Study of the information needed by users of the system and of the various performance requirements relative to those needs. The preliminary design provides the user with an overview of the system.
General design	General design report	A design of the overall system, defining input and output formats and specifying the flow of work and major requirements, including files to be maintained, major processes to be performed, and control requirements to be achieved.
Detail design	Detail design report	The complete specifications for all files and records, program modules, and procedure manuals, as well as plans for testing, conversion, and training.
Program development	Tested programs	Detailed program planning, coding, and testing.
Procedure development	User manuals and operator manuals	The writing of procedures and testing for clarity and completeness.
Conversion	Conversion report	Preparation or conversion of files, training, and the switchover to the new system.
Post implementation review	Report on the system	A review of the system in use and an evaluation of its performance.

**Table 4-1** Application development life cycle

## **USER AND AUDITOR INVOLVEMENT IN THE APPLICATION DEVELOPMENT LIFE CYCLE**

Since user requirements should be the main reason for developing an application system, the users should be involved in every phase of system development. To document this involvement, user approval should be required at the end of each development phase. In a large project designed by a development team, a user should be assigned as a team member and may be assigned as project leader.

There are two views regarding the role of the auditor in application system design. One view is that the auditor should not be involved in application system design. Auditors must make an objective review of the system. If they become involved in the design of the system, they can no longer be objective since they would have to review their own work. According to this view, the auditor should do a review only after the application is completed.

The other point of view acknowledges this potential conflict of interest but is more concerned with the difficulty and cost of making major revisions in an EDP system after it is designed. A control feature that could have been easily incorporated in the original design of a system may be too costly or take too much time to install after a system has been established. Because an auditor is an expert in the area of accounting controls, he should logically participate in the original specification and design of such controls. However, the auditor should only be concerned with the design of controls; involvement in the implementation or operation of an application system is not part of the audit function.

Both approaches acknowledge the contribution of the auditor in the area of system controls. Both approaches call for the auditor to offer suggestions to correct control weaknesses. They differ only in the relative emphasis assigned to objectivity and cost.

## **ALTERNATIVES TO TRADITIONAL DEVELOPMENT LIFE CYCLE**

The foregoing traditional life cycle is applicable to well-defined data processing applications. Most accounting applications benefit from the structure and quality assurance imposed by the life-cycle method. However, there are applications for which the traditional life cycle has not worked as well. These are typically applications for which the requirements cannot be well specified in advance, because part of the requirements become known only as the system is used. Many applications supporting planning, decision-making, and management control are of this type. An alternative to the traditional development cycle in such cases is a prototyping development approach (a heuristic or trial-and-error cycle) in which a prototype application containing initial specifications is developed and implemented, but the design and the software system used in development allow the application to be changed frequently as the system is used. When users are satisfied with the system, either it may be redone with more traditional documentation or the prototype system may continue in use.

Another development approach especially suited to small applications on time-sharing systems or on personal microcomputers is to turn the development over to the user. In many cases, the applications being designed by the user are for planning and decision-making and have a short life or are revised frequently. There is some concern among auditors that such user-developed systems are not adequately tested and controlled. Documentation, backup, operating procedures, and other controls may be inadequate if the user-developed systems are to be used as part of the formalized information processing system.

## **Testing of Application Systems During Development and Implementation**

After the programs have been written and before a system can become operational, it must be tested. Testing is a major part of the development of a program, accounting for 20 to 50 percent of the effort needed to implement an application system.

Testing of an application can take place on five different levels: program/module tests, string tests, system tests, pilot tests, and parallel tests.

### **PROGRAM/MODULE TESTS**

Each program or module is tested separately. As far as possible, these tests are designed to test the full range of the processing logic. After all the modules have been tested on an individual basis, they are combined in programs, each of which is tested on an overall basis using the same data used to test the modules. While some tests may be specified by users of the system, program/module tests are often designed by the programmer who wrote the code. Test results are reviewed by the systems designer or a programming manager; in some installations, the quality assurance group may become involved in this level of testing.

A number of software aids are available for module/program testing. Cross-reference systems can be used to identify each line of code in a program in which a particular data name is used. Some systems identify the initial source of the data contained in each field (or data name), list each reference to that field, indicate the nature of the reference (data movement, computation, update, comparison, and so on), and record the final disposition of the field (output file, report, and so on). Programmers can use other software aids to locate logic errors, trace the execution of a program step by step, and provide a picture of changes to each field during processing.

### **STRING TESTS**

Sets of related programs are called "strings." For example, a payroll system made up of fifty-six programs might contain a four-program subsystem that is used to update the master file; the four programs in this subsystem would be a string.

A string test verifies the interface between related programs. Since problems in this area are common, it is important to provide a wide range of test data for string testing. The tests should examine the validity of the data passing from program to program to determine that errors are identified at the earliest point in the processing cycle. String tests are usually prepared and conducted by the systems designer, and the results are reviewed by the project manager or persons assigned to the quality control function.

### **SYSTEM TESTS**

System tests are applied to all the programs in a system. In the example of a payroll system containing fifty-six programs, a test of all fifty-six programs would be a system test. Data are processed through all the programs and subsystems to test the interface and interactions that take place in the system. The tests are designed to simulate processing of a wide range of both valid and invalid transactions. System tests are normally designed by the systems designers and the project manager, and they are reviewed by a quality control group or the data processing manager.



**PILOT TESTS**

A pilot test involves the processing of one or more cycles of actual transactions to establish representative cycles. The results of this processing can be compared with the results obtained from processing through the existing system. Reconciliation procedures may be required if the system being developed does not produce output that can be directly compared with output from the current system. A pilot test is sometimes called a "volume test," since it involves the processing of a volume of transactions equal to the transactions in a given period. Pilot tests are usually designed by the project manager and are reviewed by the user departments or someone in a quality control position. The quality of testing is improved if the user is responsible for pilot testing.

**PARALLEL TESTS**

A parallel test involves the simultaneous operation of the old and new systems and a comparison of the resulting outputs.

Running parallel tests may require renting extra equipment, hiring temporary employees, and having current employees work overtime. User departments may have to prepare two sets of input documents, one for the old system and one for the new. Special programs to compare the output files from one system with those from another may have to be written and tested. These activities require considerable effort, but parallel testing is vital for complex systems, for example, on-line file updates.

Parallel tests should be jointly planned by the project manager, EDP operations, and the users, and the test results should be reviewed and evaluated by these groups.

**AUDITOR'S ROLE IN TESTING**

Auditors should review the test data up through system test and add any additional tests that are required for audit purposes. By a review of the results obtained from processing these expanded tests, the auditor can evaluate both the controls and the effectiveness testing. Having tested the system at this point, the auditor will probably choose to follow up with a review of the pilot and parallel test results.

**Relationship of Organization and Management Practices to Accounting Controls**

This chapter emphasizes good organization and management practices for EDP, without distinction between practices that provide accounting control and those that provide administrative control. The following list identifies accounting controls within the scope of the audit and distinguishes them from administrative controls.

Organization and Control Practice

- Position descriptions
- Segregation of functions within EDP
- Segregation of user and EDP functions

Classification

- Administrative
- Accounting and administrative
- Accounting

(continued)

Organization and Control Practice (cont )

Classification (cont )

Control functions	
User control	Accounting
Processing control	Accounting
Librarian	Accounting and administrative
Data administration	Accounting and administrative
Selection and scheduling of personnel	Administrative
Standard operating procedures	Accounting and administrative
Standards	Administrative
Master development plan	Administrative
Application development life cycle	Accounting and administrative
User involvement	Accounting
Auditor role	Accounting
Testing of applications	Accounting and administrative
User participation	Accounting
Auditor role	Accounting

The accounting control elements of these organizational and control practices are summarized below

*Segregation of functions within EDP*

- Segregation of functions of development, operations, and control
- Rotation of operators and program maintenance personnel
- Vacations by EDP employees

*Segregation of user and EDP functions*

- Origination, authorization, or correction of transactions
- Custody or control of non-EDP assets
- Authorization to change EDP controls
- Origination of master file changes

*User control*

- Independent control over input transactions
- Review of transactions and file changes processed

*Processing control*

- Review of data to be processed
- Reconciliation of controls developed in processing with input control information
- Correction resubmission control
- Output distribution control

*Librarian function*

- Custody of documentation and data and program files
- Control over access to documentation and files

*Data administration function*

- Control over access to database
- Control over database changes

*Standard operating procedures*

- Computer operator instructions, including errors and control instructions

*Application development life cycle*

- Active participation by users (and internal auditors as appropriate)
- Management review and approval at each phase of development life cycle
- Independent review and evaluation of systems during development life cycle

*Testing of applications*

- Participation of users in testing new applications for conformity with requirements and specifications

## **Summary**

This chapter describes some of the preferred practices that have been applied to the organization and management of the EDP function. The plan of organization for the data processing area should emphasize a proper segregation of functions and the effective implementation of a formal control function.

In the management of a computer installation, three particular areas should receive special consideration: organizational and personnel practices, the use of standard operating procedures, and adherence to data processing policies and procedures. The evolution of the EDP system should be governed by a master development plan. The plan should cover planned activity in the areas of application systems, system software, equipment, and personnel.

Proper management and control of application systems development is also necessary. The development process involves distinct phases with reviews and products at the end of each phase. Quality assurance for programming may use a variety of methods, including modular program development. In addition, a successful development project usually requires both user and auditor involvement.

Application development testing is also an important management and control requirement. Five levels of testing should be provided: program/module, string, system, pilot, and parallel. The user's role in these test activities is an essential control element.



---

## Documentation of the Application

# 5

Documentation consists of records, reports, workpapers, and other materials that describe the system and procedures for performing data processing tasks. It is a means of communicating the essential elements of the data processing system, including the logic followed by the computer programs. Preparing documentation is a necessary, though frequently neglected, phase of data processing. Documentation can—

- Provide for management review of proposed application systems.
- Provide explanatory material for users
- Instruct new personnel by providing background on previous application systems and by serving as a guideline for developing new applications
- Provide the data necessary for answering inquiries about the operation of a computer application
- Serve as one source of information for an evaluation of internal control
- Provide operating instructions
- Simplify program revision by providing details of processing logic
- Supply basic information for planning and implementing audit software or other auditing techniques

When documentation has not been maintained, users have found modification of an existing system to be very expensive. For example, a change in withholding rates or the addition of a city income tax has, in some cases, necessitated a substantial amount of program revision because documentation did not contain the information needed to make the required modification. Documentation is especially important when programs are coded in low-level, machine-oriented languages rather than in high-level languages such as COBOL, which are partially self-documenting.

## Types of Documentation

An EDP application system, such as payroll, may include a number of processing tasks that are performed by separate computer programs. Although the programs are separate entities, they are often interconnected since the output from one program may be input to another. An interconnected set of programs is usually called a "job." For example, the sample documentation included in Appendix C of this book is a portion of an inventory system related to the job of performing daily updates of the master file. The job contains programs to create and to update an inventory master file.

The documentation describes the relationships of the jobs in the application system, the programs in the application job, and the logic of each program. The basic types of documentation include the following:

- System documentation
- Job documentation
- Program documentation
- File documentation
- Operations documentation
- User documentation

Note the separation of documentation for programs and files. At one time, files were documented as part of each program. This was a valid approach when each program had its own unique files, however, the sharing of integrated files by a large number of programs makes it more practical to prepare separate documentation for each file and to reference that documentation to each program using the file. This approach simplifies the updating of file documentation.

### SYSTEM DOCUMENTATION

System documentation represents the highest level of documentation for an application, and, as a result, it usually contains the least amount of detail. Typical application system documentation contains a narrative describing the system, a flowchart showing the jobs and programs involved, a flow diagram describing the clerical processing required, a report describing in detail all input documents and outputs, and a list containing the control functions and operations built into the system.

This documentation is designed to present an overall view of a particular application system. It provides information for management and users of the system, and its documentation of controls is of particular interest to auditors because it provides a good starting point for the study and evaluation of accounting and administrative controls.

### JOB DOCUMENTATION

Job documentation normally contains the following elements:

#### Job Narrative

This is a statement of the purpose of the job plus any unusual processing requirements.

### **Run-to-Run Flowchart**

This illustrates the program-to-program flow of all input to and output from the set of programs used in the job. The source of all input and the final disposition of all input and output are clearly identified. The flowchart provides an overview of a particular job.

### **Program Summary**

The summary contains a list of the programs used in the job. Programs are identified by reference number (usually a six- or eight-character name used to refer to the program in the job command statements and for internal purposes) and by descriptive name. For each program, run frequency is shown, along with the estimated running time for normal and peak processing requirements.

### **Source Document Summary**

This summary lists, by name and reference number, all input to the job. The summary provides an estimate of average volume and references each transaction to the file that it updates or creates. Other special characteristics, such as transaction codes or optical scanning requirements, are noted when applicable. (Detailed descriptions of each input screen or source document are included as part of the system documentation section.)

### **File Summary**

This lists each file used in the job, supplying the following information:

- Identification number (usually a six- or eight-character name that will be used by the operating system)
- Usage (input, output, or work file)
- Medium (card, tape, disk, and so on)
- Source (input source transactions or other files used to update or create this file)
- Volume (estimated transactions or records)
- Retention period

This summary provides an overview of the file requirements for a particular job and the basic cross-reference (identification number) that is used to locate the detailed documentation for the file.

### **Output Summary**

This summary lists each report or output screen produced by the job, supplying the following information:

- Identification number
- Identification number of the program that produces the report or screen
- Frequency
- Form number of the paper on which the report is printed
- Estimated average report volume in terms of print lines
- Report or screen format

This summary provides a cross-reference between jobs, reports, and programs.

## 62 Documentation of the Application

### **PROGRAM DOCUMENTATION**

Intended to serve as the basis for inquiries and program and system revisions, this documentation contains detailed information about each program .

#### **Program Narrative**

The program narrative presents an overview of the program and typically includes a description of the purposes of the program, the inputs and outputs, and the processing requirements. The program narrative is useful both as a description of the program for a nontechnical reviewer and as an introduction to the detailed documentation to follow.

#### **Program Structure and Logic Description**

Traditional documentation of program structure and logic is the program flowchart. Many installations do not use them because they are difficult to prepare and keep current. Some installations use only well-documented and well-structured program code. Other installations use supplementary documentation such as hierarchy charts to show program module structure, HIPO (hierarchy, input, processing, and output) charts to show structure and logic flow, and English-like pseudocode descriptions of program logic.

#### **Source Code Listing**

A key documentation element is a listing of the program source code. The program should be coded so the program logic can be clearly understood. If COBOL is used and reasonable coding standards are enforced, the source code listing is informative. In other languages, it may be necessary to include comments to explain the program.

#### **Test Results or Sample Output**

Ideally, a program should be subject to a comprehensive test based on a set of data specifically designed to execute the program logic. The basic set of comprehensive test data for an application is often called base case test data. The test data and the test output should be retained in the documentation. It can be used in the review of output from the processing of the same test data following a subsequent revision.

#### **Change Notice**

Each time a program is modified, a change notice should be prepared and added to the program documentation. (See figure 5-1.) These notices provide a complete log of all revisions. The notice contains information such as the following:

- A statement of the purpose of the revision
- A description of the revised processing
- A list of the detail changes that were made to the program source code
- An indication of when the change was tested and implemented
- A sign-off indicating supervisory review and approval



<b>Program Change</b>		
Program name _____	Application _____	
Date requested _____	Date needed _____	
Requested by _____	Department _____	
Change request approved by data processing _____	Date Approved _____	
Purpose of the revision		
_____		
_____		
_____		
Description of revised processing		
_____		
_____		
_____		
List of changes made		
<i>Module/documentation</i>	<i>Changes</i>	
_____	_____	
_____	_____	
_____	_____	
Change tested by _____	Date _____	Estimated cost \$ _____
Change approved by DP _____	Date _____	Actual cost \$ _____
Change approved by user _____	Date _____	Estimated hrs _____
Change effective on (date) _____		Actual hrs _____
Fallback procedure to be followed in the event of difficulties with change _____		
_____		

**Figure 5-1** Program change form

## FILE DOCUMENTATION

The use of integrated files and databases makes it logical and practical to prepare separate file documentation and reference it to all the programs that use each file. In COBOL programs, the FD (file description) entry from the source code may, with the addition of suitable comments, provide adequate documentation. If more than one COBOL program uses a file, separate file documentation can be achieved by preparing a single file description stored in a COBOL library of such descriptions. Each program obtains the file description by using a COPY instruction that causes the file description from the library to be incorporated into the COBOL program.

A separate type of documentation, called a "data dictionary," can be used to record information about the structure and content of all computer-based files. The data dictionary is a fundamental EDP management tool. It is based on the concept that a set of data is a resource that should be shared by all the application users. To permit this sharing, the data itself must be consistently defined. A data dictionary is usually implemented in conjunction with a database management system, however, it has also been used to systematize the data definitions in complex, batch-oriented computer systems. The data dictionary approach is illustrated in figures 5-2 and 5-3.

The data dictionary provides the following documentation:

- Standard definitions for all data elements, records, and files
- Narrative and technical descriptions of security provisions, edit considerations, logical and physical structures, and applications use of all data
- Cross-references for data element creation and use

This documentation approach offers the following advantages:

- It allows strict control over the definitions assigned to each element within the system
- It simplifies the tracing of the path of data items through a system
- It encourages the use of standard edit and validation routines so that the same types of validation will be applied to a data element no matter where it enters the system
- It becomes easier to evaluate the impact of a change on the existing data structure, since all elements in the structure are cross-referenced to their source and use
- It reduces the applications program effort by providing standard definitions that may be used by everyone involved in systems development
- It serves as the initial step in the design and implementation of a database system

However, preparation of a data dictionary is complex and time consuming. In a medium-scale system, it might take one or two work-years to create a data dictionary. The continued support of the data dictionary requires a heavy maintenance effort. Once a data dictionary has been established, formal control must be provided over the setup of, and access to, all new files that are created. Often, this type of control is difficult to implement in a small- or medium-scale installation.

In most cases, the advantages inherent in a data dictionary outweigh the problems, and a trend toward the use of this documentation technique has developed.

	Document Type Segment Description	Revision No. 0	Page 1	Of 1
System Parts	Name VENDRSEG	Revision Date	Original Date 3/24/8X	
Subsystem Vendor Information	Number PS30000	Analyst DLA	Design Level	
			General	Detail
				X
<p><u>Description</u></p> <p>This segment contains information about vendors who are possible sources of supply for a particular part. If a part is not purchased from the outside (e.g., internally manufactured), this segment will not be present. When it is present, this may be a repeating segment.</p> <p><u>Parent Segment:</u> PARTROOT</p> <p><u>Dependent Segment:</u> VADDRSEG</p> <p><u>Data Elements</u></p> <p>VENDR-ID-CODE  VENDR-NAME  VENDR-LAST-PRICE  VENDR-LAST-QTY-ORDERED  VENDR-LAST-ORDER-DATE  VENDR-LAST-ORDER-PO-NO</p>				

Figure 5-2 Data dictionary segment description

	Document Type Data Element Description	Revision No. 0	Page 1	Of 1
System Parts	Name VENDRSEG	Revision Date	Original Date 3/24/8X	
Subsystem Vendor Information	Number PS300030	Analyst DLA	Design Level General Detail x	
<p><u>Element:</u> VENDR-LAST-PRICE</p> <p><u>Name:</u> Last Price Charged by a Vendor</p> <p><u>Description:</u></p> <p>This element contains the most recent price a particular vendor charged, per unit of measure, for a particular part.</p> <p><u>Physical Attributes:</u> PICTURE 9(4)V9(2).</p> <p><u>Values:</u> Positive values from 0000.01 to 9999.99</p> <p><u>Validation Considerations:</u></p> <p>The element should always be all numeric. Presently, no known value exceeds 2000.00 and it is not likely that this upper limit will ever be exceeded. An updated value placed in this element should not vary from the prior value by more than 10%. This element should not vary by more than 20% from the value of MASTER-UNIT-PRICE in the PARTROOT. A transaction with a non-numeric value in this element should be rejected. All other transactions that fail to meet the validation criteria should be held on a validation suspense file, reported on a validation exception report, and processed in accord with instructions issued by someone who knows the proper supervisory override.</p> <p><u>Owner:</u> Authorized users in the Purchasing Department</p> <p><u>Security:</u> Access restricted to authorized users in accounting, auditing, or purchasing. This element should be considered COMPANY CONFIDENTIAL.</p> <p><u>Standard Report Heading and Output Format:</u></p> <p>bbVENDORbb LAST PRICE bZ,ZZ9.99b</p> <p><u>Outputs:</u> Report/File INR237 INF393 INF407 INF632 Program INP391 INP391 INP277 INF892</p> <p><u>Sources:</u> Input/File PUR232 Program INP391</p> <p><u>Parent Segment:</u> PARTROOT <u>Dependent Segment:</u> VADDRSEG</p>				

Figure 5-3 Data dictionary data element description

To make the data dictionary as useful as possible to the widest audience, it is necessary to prepare a detailed index of the documentation contents. A number of different indexes are normally supplied. Figure 5-4 illustrates the different types of indexes that might be required.

## OPERATIONS DOCUMENTATION

Operations documentation is a special-purpose subset of job and program documentation that contains information that operations personnel need to perform their tasks. The concept of segregation of functions in data processing suggests that operators should not have access to complete documentation. Without such access, a computer operator would find it difficult to alter a program. Therefore, the computer operator should be given access only to the operating instructions, since they provide all the information necessary for running the program. Operations documentation often includes the following elements:

- *Job and program narratives* Edited versions of the narrative, outlined as part of program documentation. They do not contain the narrative description of program logic.
- *Input/output chart* Displays all the input and output files required for a computer program. The source of all input and the disposition of all output are indicated. Operations personnel use this chart to determine the files that are needed for each program and to determine how output files and other output are to be distributed or handled when processing is complete.
- *Job command language statement list* A list of the job command language statements that are required to execute a job or an individual program. If there is a problem with the processing of a program, this list is checked to determine if the program has been properly set up for operation. Further, if the control statements for the command language become lost, this documentation is the source of the information needed to reconstruct them.
- *Error message list* Provides the operator with a list of all the messages contained in the program. The meaning of each message is explained, and the operator is given a description of the required action to be taken in response to the messages.
- *Restart and recovery instructions* Provides detailed instructions about the restart and recovery procedures to be employed.

## USER DOCUMENTATION

The final category of documentation is prepared for the user departments. This contains a nontechnical description of the system, the input, and the output.

Control procedures are listed in detail and include a list of the people responsible for performing each of these procedures. These responsibilities should be identified by position or department, not by the name of the individual assigned, since personnel may change. Error correction procedures, any cut-off procedures for the submission of input data, and any special timing considerations involved in processing should be clearly indicated.

The user manual for on-line applications should include instructions and explanations for—

- Logging on the terminal and the application, including security procedures and warnings.

Hierarchical Index

PART DATA BASE (cont.)	Page
VENDRSEG	62
VENDR-ID-CODE	63
VENDR-NAME	64
VENDR-LAST-PRICE	65
VENDR-LAST-QTY-ORDERED	66
VENDR-LAST-ORDER-DATE	67
VENDR-LAST-ORDER-YR	68
VENDR-LAST-ORDER-MONTH	69
VENDR-LAST-ORDER-DAY	70
VENDR-LAST-ORDER-PO-NO	71

Alphabetic Index

	Page
VENDR-ID-CODE	63
VENDR-LAST-ORDER-DATE	67
VENDR-LAST-ORDER-DAY	70
VENDR-LAST-ORDER-MONTH	69
VENDR-LAST-ORDER-PO-NO	71
VENDR-LAST-ORDER-YR	68
VENDR-LAST-PRICE	65
VENDR-LAST-QTY-ORDERED	66
VENDR-NAME	64
VENDRSEG	62

KWIC (Keyword in Context) Index

	Page
ORDER - (VENDR-LAST-ORDER-DATE)	67
ORDER - (VENDR-LAST-ORDER-DAY)	70
ORDER - (VENDR-LAST-ORDER-MONTH)	69
ORDER - (VENDR-LAST-ORDER-PO-NO)	71
ORDER - (VENDR-LAST-ORDER-YR)	68
ORDERED - (VENDR-LAST-QTY-ORDERED)	66

Figure 5-4 Data dictionary indexes

- Help or training aids and expert mode
- The various menus that may be provided to guide the operator
- Data fields to be entered
- Error messages and reentry procedures
- Recovery and restoration procedures if an on-line session is interrupted
- Log-off and end-of-day procedures

The documentation should also include a description of how the user departments should determine the accuracy of the output they receive. Auditors may find it useful to review user documentation to obtain a general understanding of the functions performed by the user and to assist in developing an understanding of the flow of information between the user and EDP.

## Software Documentation Aids

One of the major obstacles to the preparation of adequate documentation is the amount of time required to develop and maintain it. This problem has been partially solved by the development of software packages that can reduce the manual effort involved in documentation. These packages may be classified as follows:

- Flowchart packages
- Cross-reference listings
- File description generators
- Formatting or shorthand systems
- Librarian packages

### FLOWCHART PACKAGES

The flowchart packages read the source code for a program and convert it into a flowchart, which is printed on the line printer. Boxes and lines are drawn by the use of symbols that are part of the printer's normal character set, such as an asterisk, a dash, and so on. Even with the graphic limitations imposed by the use of these symbols, computer-generated flowcharts are quite readable. They are larger than hand-drawn flowcharts, since there is no combining of like operations. Figure 5-5 is a portion of the flowchart produced by using a package to process the source code from the program in this book's Appendix C.

The majority of the flowchart packages are designed to process COBOL source code, but some systems can handle FORTRAN, PL/1, or ASSEMBLER language. The detailed information or comments printed within the blocks on the flowchart will be based solely on what appears in the source code. If the program logic is intricate and the data names are cryptic, the flowchart will be difficult to follow; if the logic flow is well planned and the data names are descriptive, the flowchart will be clear and easy to understand.

In general, flowchart packages have the following advantages:

- The same information or situation is always charted in the same way
- The flowchart accurately reflects the current status of the source code





INFO82	/021010/	020140	100050	100080	140230
INFO83	/020010/	020090	100040	140220	260040
INVENTORY-MASTER-RECORD	/020040/	290030			
INVENTORY-TRANSACTION-RECORD	/021040/				
MAINLINE	/100060/	100150	100180	100210	100230
MASTER-FILE-INDICATOR	/059020/	100130	105025	260050	
MASTER-LAST-RECEIVED-DATE	/020130/	110040			
MASTER-LAST-RECVD-DAY	/020160/				
MASTER-LAST-RECVD-MONTH	/020150/				
MASTER-LAST-RECVD-YR	/020140/				
MASTER-LAST-USAGE-DATE	/020090/	120110			
MASTER-LAST-USE-DAY	/020120/				
MASTER-LAST-USE-MONTH	/020110/				
MASTER-LAST-USE-YR	/020100/				
MASTER-PART-DESCRIPTION	/020060/				
MASTER-PART-NUMBER	/020050/				
MASTER-QTY-ON-HAND	/020080/	110030	120030	120050	120100
MASTER-UNIT-PRICE	/020070/				
MOVE-TRANSACTION-TO-OUTPUT	/160010/	105040	120060	130050	300060

**Figure 5-6** Sample cross-reference report

## 72 Documentation of the Application

- There is no need to store and maintain bulky flowcharts since a new chart can be prepared whenever one is needed
- The charts can be prepared by the computer at a lower cost than the equivalent charts prepared manually

Auditors frequently encounter systems that are not well documented. When reviewing the system and, in particular, the detailed aspects of some internal controls, the auditor can find a flowchart quite helpful. The auditor can use the flowchart package to produce documentation that will provide a starting point for further review.

### **CROSS-REFERENCE LISTINGS**

Cross-reference listings provide an alphabetic list of all names used in a program, indicate the line number in the source code where the name was first defined, and list all other line numbers that reference that name. Program compilers (and some flowchart packages) produce a cross-reference list as a byproduct of their normal operations, while others offer such lists as an option. The cross-reference lists are useful in reviewing the logic of a computer program. Figure 5-6 shows a portion of the cross-reference report for the program documented in Appendix C.

### **FILE DESCRIPTION GENERATORS**

To understand a computer program or system, the auditor must learn the structure of the files that are processed. A section of the program is devoted to a definition of the files that are used. For the most part, the formats of these definitions are difficult for someone other than a programmer to understand.

A number of programs have been developed to read the descriptions as they appear in the computer program and to convert them into a graphic or tabular format that is more easily understood by the nontechnician. These programs can be quite helpful in providing supplementary documentation. Figure 5-7 shows an example of the output produced by one of these software packages.

### **FORMATTING**

Coding format standards make it considerably easier for one programmer to read the work of another, however, it is often difficult to enforce standards because of differences in training among programmers and because of a common feeling that format standards stifle creativity. To achieve standardization, some installations have made use of the formatting capabilities available in some preprocessor software systems. These preprocessors accept a source code written in any format and convert it to a standard format established by a particular installation.

The preprocessor packages also allow the programmer to write a shorthand version of the language. The formatting program converts the shorthand into its corresponding full-text version. A user table for the conversions of data names must be provided. Standard language elements, such as the COBOL picture clause, are based

```

PROGRAM G555
INPUT RECORD LAYOUT
*****MASTR*****
**TYPE 1**
/MATCH *M1*/
---/CLTNO/ . . . /BRANCH/ . . . /ERRORS/CLOSE/SMONTH/. . . /YEAR/CLTNME/HDATE/---
  3  6  12 13 13 20 20 21 21 22 23 26 27 28 59 60 77
                                     /SDATE--- --- /
**TYPE 2**
/MATCH *M1* --- /
---/CLTNO/ACTN01/ACTN02/ . . . /TCODE1/TCODE2/ . . . /DESCPT/ . . . /PCODE/---
  3  6  7  9 10 12 20 20 21 21 28 59 78 79
**TYPE 3**
/MATCH *M1* --- /
---/CLTNO/ACTN01/ACTN02/ . . . /TCODE1/TCODE2/ . . . /DESCPT/ . . . /YTDAMT *P*/ADJ *P*/
  3  6  7  9 10 12 20 20 21 21 28 59 66 71 72 77
**NOT SHOWN ABOVE **
---/CUR *P*/---
  59 65
**TYPE 4**
/MATCH *M1* --- /
---/CLTNO/ACTN01/ACTN02/ . . . /TCODE1/TCODE2/ . . . /DESCPT/ . . . /PCODE/---
  3  6  7  9 10 12 20 20 21 21 28 59 78 79

```

Figure 5-7 Computer-generated file description

SEQUENCE		A	B	COBOL STATEMENT												IDENTIFICATION								
1	3	4	6	7	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	
					FD		INF082	3C	10	DR	I-T-REC.													
					01		I-T-REC.																	
							05	T-P-NUM	P9999.															
							05	T-T-COD	P9.															
							05	T-QTY	PIC9999999.															
							05	T-DAT.																
							10	T-MON	P99.															
							10	T-DAY	99.															
							10	T-YER	P99.															

Figure 5-8 Programmer-written code using abbreviations

on standard abbreviations provided by the package. Figure 5-8 illustrates the coding that was submitted by a programmer, and figure 5-9 shows the code generated by one of these format/shorthand packages.

## LIBRARIAN PACKAGES

Over the years, a number of different librarian software packages have been developed to help maintain computer software. These packages enable an EDP department to establish a library containing a copy of each source program. There are procedures to update, delete, and change the programs contained in that library. It makes it considerably easier to maintain the system, since the librarian system provides facilities to alter programs easily whenever a change is required. The librarian packages generally provide the capability for maintaining a log that keeps track of all changes that are made to each program; this log can be an invaluable aid in tracing such changes. Further, some of the librarian packages provide a limited formatting capability that can be applied to the programs that are written.

The librarian packages may also be useful in enforcing other standards within an installation. For example, particular commands that are banned in a particular installation may be identified to the librarian package. Once these commands are identified, the package will not accept any program instruction containing those verbs. A common example of this might be an organization that forbids the use of the COBOL command "ALTER." Even though it is quite powerful, that particular verb can cause difficult-to-locate errors within programs, and some installations have made the decision to ban its use.

Librarian packages differ in the control over object (machine language) code. Many widely used librarian packages provide control over source code, for example, COBOL, but do not control the object code produced when the source code is compiled. Since the actual processing is based on the object code, control of source code is one step removed from the actual processing. If a change is made directly in machine language or by overlaying the object code, the source code control will not detect or record the change. There are librarian packages that do keep track of object code, which provides a major improvement in the level of program control that can be maintained within a data processing installation. Librarian systems often provide archiving capabilities to allow the recovery of prior program revisions.

Figure 5-10 shows a portion of the master index produced by one librarian package. The index is in sequence by module (program) name. For each purpose, the index reports the following:

- Password required to access the module (For security reasons, however, it is *not* a good idea to list this password on the index.)
- A brief description of the module
- The date the module was initially placed in the library
- The date and time of the last module update
- The number of lines in the module

## Minimum Documentation and the Auditor

No generally accepted minimum standards for documentation exist, either inside or outside the data processing community. However, any organization that files a tax return with the Internal Revenue Service must meet the minimum documentation standards described in Revenue Procedure 64-12 and Revenue Ruling 71-20. These pronouncements establish guidelines for the kind of documentation that must be main-

```

FD  INF002
    BLOCK CONTAINS 10 RECORDS
    DATA RECORD IS INVENTORY-TRANSACTION-RECORD.
01  INVENTORY-TRANSACTION-RECORD.
    05  TRANSACTION-PART-NUMBER          PICTURE 9{4}.
    05  TRANSACTION-TYPE-CODE          PICTURE 9{1}.
    05  TRANSACTION-QUANTITY          PICTURE 9{6}.
    05  TRANSACTION-DATE.
        10  TRANSACTION-MONTH          PICTURE 9{2}.
        10  TRANSACTION-DAY           PICTURE 9{2}.
        10  TRANSACTION-YEAR          PICTURE 9{2}.

```

**Figure 5-9** Code formatted by the formatting package

## MASTER LIBRARY INDEX

05/09/8X      TIME 1406      PAGE 37

MODULE	PASSWORD	MODULE DESCRIPTION	ADDED	LAST UPDATE	RECORDS
INP040	GACB	INVENTORY TURNOVER	09/01/8X	09/23/8X 1240	437
INP041	RUSD	INVENTORY ACTIVITY	10/10/8X	12/17/8X 0831	683
INP042	TMNO	UPDATE USAGE FILE	11/20/8X	01/13/8X 0914	237
INP043	FRIC	USAGE DETAIL LIST	01/30/8X	NONE	743
INP060	QUES	UPDATE MASTER DESC	03/03/8X	05/09/8X 1201	1064
INP062	ROUW	PRINT MASTER LIST	08/07/8X	05/09/8X 1401	384
INP063	MDDL	RECREATE MASTER	07/05/8X	NONE	443
INP081	SCLJ	TRANS CARD TO TAPE	03/08/8X	03/22/8X 1837	71
INP082	LLMR	UPDATE INVNTY MAST	04/01/8X	05/08/8X 1327	562
INP083	RFDS	CREATE BACKUP	04/09/8X	NONE	486
INP100	RFCU	YEARLY ACTIVITY	02/09/8X	02/11/8X 1304	749
INP101	NCYM	YEARLY SUMMARY	03/31/8X	09/28/8X 1743	868
LGP010	OKLR	WRITE COMM LOG	05/24/8X	05/26/8X 1801	1437
LGP020	EOUV	PRINT LOG REPORT	11/29/8X	12/24/8X 1409	1141
LGP030	AED1	UPDATE CUM LOG	12/31/8X	03/27/8X 0940	2349

**Figure 5-10** Master library index

tained in support of a computerized application that has an impact on a filing with the Internal Revenue Service. Copies of the procedure and the ruling appear in figures 5-11 and 5-12 at the end of this chapter.

Minimum documentation standards have not been developed and accepted by the data processing community because there are significant differences among installations in the approach to EDP and systems design. As a result, acceptable documentation may take many different forms. However, it is possible to establish general guidelines for the content of an acceptable documentation plan. Basically, three types of documentation should be prepared: (1) systems documentation, (2) program documentation, and (3) operations documentation. Minimum documentation requirements are outlined for each of the three types. This is the minimum level of documentation an auditor should generally expect to find in any system he or she is engaged to audit.

## SYSTEMS DOCUMENTATION

Minimum application systems documentation normally includes a description of the reasons for implementing the application and an outline of the operations to be performed. It contains project documentation, such as copies of all project proposals or

## 78 Documentation of the Application

system outlines. Evidence that the system had been approved for development and implementation is included. A list of assigned project development responsibilities may also be provided.

At the next level, application systems documentation usually includes a narrative description of the application system and its expected processing results, plus system flowcharts illustrating the flow of transactions or data through the system and the relationships among the various processing steps and computer programs.

There should be descriptions of the input to and output from the application system, as well as descriptions and layouts of the files that are used in all stages of processing. As mentioned earlier, this kind of information might be part of a separate set of file documentation. Normally, the auditor should expect this level of documentation to contain a description of the control features incorporated into the system and an explanation of how these controls ensure the accuracy and integrity of the processing. This level of documentation should also include copies of properly approved change authorizations for each system modification.

The auditor would rely on this level of documentation in tracing data from original entry to final output or other disposition. Therefore, this documentation gives the auditor a clear indication of the processing or audit trail for the application.

### **PROGRAM DOCUMENTATION**

Program documentation should include a narrative description of the program, plus a copy of the program source code and a listing of all parameters or job control statements required to process the program. It may also include flowcharts, decision tables, or logic narratives to explain the details of the processing.

If they are not covered elsewhere, a list of programmed control features should be included in the program documentation. Tables may be used to explain any code values used in the program. These tables should include an explanation of the meaning and content of the codes. There should be a record of each change that has been made to the program and an indication of its effective date and proper approval. There should be copies of the formats of any outputs from the program, some indication of the operating instructions required to execute the program, and a description of any special features, such as error detection routines or the use of parameter switches implemented in the program code.

### **OPERATIONS DOCUMENTATION**

This documentation is prepared to assist operations personnel in properly executing the programs. For each program, this documentation should include a brief description of the processing and a list of the required input and output. The operator should also be informed of any special instructions regarding the setup of the computer to do the required processing. Operating system requirements are listed in terms of job control language specifications. The documentation should include a list of all program messages and halts that may be encountered during the execution of a particular program, and it should inform the operator of the action to be taken in response to each of these situations. If operations personnel are required to perform any control procedures during the processing, these should be documented. Recovery and restart procedures should be outlined, too. Estimated normal and maximum run times are included to give the operations personnel an indication of how to schedule the processing. Finally, emergency instructions regarding file retention or other considerations should be included.



The auditor can review operations documentation to understand the functions performed by the operations group. This documentation also helps the auditor plan any work on the computer system for a particular program or for the use of computer-assisted audit techniques.

Unless separate user documentation is prepared, the user's duties and responsibilities should be included as part of the operations documentation.

## THE IMPORTANCE OF DOCUMENTATION FOR AUDITS

The auditor may find it necessary to use the documentation in several ways. Two uses involve the review of internal control and the planning of audit procedures using the computer.

In conducting the study and evaluation of internal accounting control in EDP systems, documentation is important as an element of accounting control and as a source of information. The AICPA Audit and Accounting Guide, *The Auditor's Study and Evaluation of Internal Control in EDP Systems*, identifies the existence of appropriate levels of documentation as one of the general accounting controls that the auditor evaluates in an EDP installation. Application documentation is frequently the best source of information on control features (including the computer programs). Therefore, a review of the controls may depend, in part, on adequate documentation in support of the application. To some extent, the auditor may find it necessary to review systems and user documentation to increase his understanding of the control features inherent in the system.

Even if documentation is not required to support the auditor's review of internal accounting control, the absence of adequate documentation often indicates a lack of administrative control and may influence the auditor's evaluation of overall internal accounting control.

Documentation is also important in the planning of audit procedures using the computer. If the review of internal accounting control and the results of compliance testing indicate that the auditor should consider the use of computer-assisted audit techniques, details of records, formats, layouts, code structures, and other basic data are required to plan and execute the procedures. This information should be contained in the documentation. Good documentation thus serves to reduce the time an auditor needs to develop and implement computer-assisted audit procedures.

## Summary

Documentation is an essential part of any data processing system. It serves as the basis for providing information about the various components of an application system and its processing logic.

The manual preparation of documentation can be slow and expensive. As a result, several software packages have been developed to reduce the manual effort involved in creating and maintaining documentation.

Although some IRS pronouncements have established certain broad de facto standards for taxpayers, no generally recognized standards have been set forth for minimum documentation requirements. The chapter defines some general guidelines for the minimum level of documentation an auditor should expect to find in any system he audits.

**Figure 5-11** IRS Revenue Procedure 64-12

26 CFR 601.105: Examination of returns and claims for refund, credit or abatement; determination of correct tax liability.

Rev. Proc. 64-12

(Also Part I, Section 6001; 1.6001-1.)

Guidelines for record requirements to be followed in cases where part or all of the accounting records are maintained within automatic data processing systems

**SEC. 1. PURPOSE.**

.01 The purpose of this Revenue Procedure is to set forth guidelines specifying the basic record requirements which the Internal Revenue Service considers to be essential in cases where a taxpayer's records are maintained within an automatic data processing (ADP) system. References here to ADP systems include all accounting systems which process all or part of a taxpayer's transactions, records, or data by other than manual methods.

.02 The technology of automatic data processing is evolving rapidly; new methods and techniques are constantly being devised and adopted. Accordingly, the five points set forth in section 4 of this Revenue Procedure are not intended to restrict or prevent taxpayers from obtaining the maximum benefits of ADP provided the appropriate information is present or can be produced by the system. These guidelines will be modified and amended as the need indicates to keep pace with developments in automatic data processing systems.

**SEC. 2. BACKGROUND.**

The inherent nature of ADP is such that it may not be possible to trace transactions from source documents to end results or to reconstruct a given account unless the system is designed to provide audit trails. Taxpayers already using ADP or contemplating its use have requested information concerning the types of records that should be developed and maintained in order to meet the requirements of section 6001 of the Internal Revenue Code of 1954 and the corresponding regulations. This section of the Code reads in part as follows:

Every person liable for any tax imposed by this title, or for the collection thereof, shall keep such records, render such statements, make such returns, and comply with such rules and regulations as the Secretary or his delegate may from time to time prescribe

**SEC. 3. OBJECTIVES.**

Modern machine accounting systems are capable of recording business transactions much more rapidly and with greater accuracy than manual systems and they are capable of retaining and producing vast amounts of data. The ability to produce in legible form the data necessary to determine at a later date whether or not the correct tax liability has been reported must be carefully considered in designing and programming a machine system. This factor may add to the complexity of the system and require additional cost, but this cost may be negligible in comparison to the expense that may be incurred at a later date if the system cannot practically and readily provide the information needed to support and substantiate the accuracy of the previously reported tax liability.

**SEC. 4. ADP RECORD GUIDELINES.**

.01 ADP accounting systems will vary, just as manual systems vary, from taxpayer to taxpayer. However, the procedures built into a computer's accounting program must include a method of producing from the punched cards or tapes visible and legible records which will provide the necessary information for the verification of the taxpayer's tax liability.

.02 In determining the adequacy of records maintained within an automatic data processing system, the Service will consider as acceptable those systems that comply with the guidelines for record requirements as follows:

(1) *General and Subsidiary Books of Account.*—A general ledger, with source references, should be written out to coincide with financial reports for tax reporting periods. In cases where subsidiary ledgers are used to support the general ledger accounts, the subsidiary ledgers should also be written out periodically.

Figure 5-11 (continued)

(2) *Supporting Documents and Audit Trail.*—The audit trail should be designed so that the details underlying the summary accounting data, such as invoices and vouchers, may be identified and made available to the Internal Revenue Service upon request.

(3) *Recorded or Reconstructible Data.*—The records must provide the opportunity to trace any transaction back to the original source or forward to a final total. If printouts are not made of transactions at the time they are processed, then the system must have the ability to reconstruct these transactions.

(4) *Data Storage Media.*—Adequate record retention facilities must be available for storing tapes and printouts as well as all applicable supporting documents. These records must be retained in accordance with the provisions of the Internal Revenue Code of 1954 and the regulations prescribed thereunder.

(5) *Program Documentation.*—A description of the ADP portion of the accounting system should be available. The statements and illustrations as to the scope of operations should be sufficiently detailed to indicate (a) the application being performed, (b) the procedures employed in each application (which, for example, might be supported by flow charts, block diagrams or other satisfactory descriptions of input or output procedures), and (c) the controls used to insure accurate and reliable processing. Important changes, together with their effective dates, should be noted in order to preserve an accurate chronological record.

#### SEC. 5. COMMENTS OR INQUIRIES.

Comments or inquiries relating to this Revenue Procedure should be addressed to the Assistant Commissioner (Compliance), Attention: CP:A, Washington, D.C., 20224.

## Figure 5-12 IRS Revenue Ruling 71-20

### Rev. Rul. 71-20<sup>1</sup>

Advice has been requested whether punched cards, magnetic tapes, disks, and other machine-sensible data media used in the automatic data processing of accounting transactions constitute records within the meaning of section 6001 of the Internal Revenue Code of 1954 and section 1.6001-1 of the Income Tax Regulations.

In the typical situation the taxpayer maintains records within his automatic data processing (ADP) system. Daily transactions are recorded on punched cards and processed by the taxpayer's computer which prints daily listings and accumulates the individual transaction records for a month's business on magnetic tapes. At the month's end the tapes are used to print out monthly journals, registers, and subsidiary ledgers and to prepare account summary totals entered on punched cards. The summary data from these cards is posted to the general ledger and a monthly printout is generated to reflect opening balances, summary total postings, and closing balances. At the year's end several closing ledger runs are made to record adjusting entries. In other situations taxpayers use punched cards, disks, or other machine-sensible data media to store accounting information.

Section 6001 of the Code provides that every person liable for any tax imposed by the Code, or for the collection thereof, shall keep such records as the Secretary of the Treasury or his delegate may from time to time prescribe.

Section 1.6001-1(a) of the Income Tax Regulations provides that any person subject to income tax shall keep such permanent books of account of records, including inventories, as are sufficient to establish the amount of gross income, deductions, credits, or other matters required to be shown by such person in any return of such tax.

Section 1.6001-1(e) of the regulations provides that the books and records required by this section shall be retained so long as the contents thereof may become material in the administration of any internal revenue law.

It is held that punched cards, magnetic tapes, disks, and other machine-sensible data media used for recording, consolidating, and summarizing accounting transactions and records within a taxpayer's automatic data processing system are records within the meaning of section 6001 of the Code and section 1.6001-1 of the regulations and are required to be retained so long as the contents may become material in the administration of any internal revenue law.

However, where punched cards are used merely as a means of input to the system and the information is duplicated on magnetic tapes, disks, or other machine-sensible records, such punched cards need not be retained.

It is recognized that ADP accounting systems will vary from taxpayer to taxpayer and, usually, will be designed to fit the specific needs of the taxpayer. Accordingly, taxpayers who are in doubt as to which records are to be retained or who desire further information should contact their District Director for assistance.

See Revenue Procedure 64-12, C.B. 1964-1 (Part 1), 672, which sets forth guidelines for keeping records within an ADP system.

<sup>1</sup>Also released as Technical Information Release 1062, dated December 31, 1970

---

## Hardware Features for Control Over Equipment Malfunctions

# 6

Modern computer equipment is very reliable, and the auditor can assume it will detect any machine-based errors. Usually, the auditor can rely on built-in hardware controls. In order to understand how the equipment works, to understand why it is reliable, to recognize the potential for errors, and to evaluate the effectiveness of machine checks, the auditor should have a general understanding of built-in hardware control features.

### How the Equipment Can Malfunction

A computer system consists of both electronic elements and mechanical parts. The central processing unit, for example, consists almost entirely of electronic elements, such as semiconductors and integrated circuits containing transistors, resistors, and diodes. On the other hand, most input/output equipment and file storage devices contain both electronic components and mechanical parts. Therefore, a failure in the system can result from the malfunction of either a mechanical part or an electronic element.

#### FAILURE IN ELECTRONIC COMPONENTS

A computer system operates by creating, counting, delaying, and transmitting electrical pulses. Communication between units of the system is by a controlled transmission of electrical pulses, and the circuitry of the computer system is designed to control their timing, shape, strength, and frequency. Failure of an electronic element, such as a transistor, resistor, or diode, may cause a change in the timing, shape, strength, or frequency of the pulses, which can lead to an error. Some of the reasons for deterioration of an electronic element are extremes of heat or humidity, power disturbances, mishandling, and normal wear.

Quality control in manufacture, built-in equipment checks, and preventive maintenance have made the electronic portion of the computer system very reliable. Preventive maintenance usually detects elements that are out of adjustment or close to failure, allowing time for adjustment or replacement. Preventive maintenance is performed daily on complex computer systems and less frequently on simple configurations. A hardware trend to include diagnostic circuits in the equipment to identify defective modules has led to more emphasis on the use of operating personnel to make minor repairs.

### **FAILURE IN MECHANICAL OPERATION**

Mechanical operation is required in almost all input/output and secondary storage equipment. Two mechanisms with mechanical actions are usually used (*a*) to move the media (input, output, or storage) past the reading or writing mechanism and (*b*) to read or write. These actions occur at high speeds. A printer that prints 1,000 lines of 136 characters per minute requires as many as 136,000 individual print mechanism movements each minute. A disk read/write head is held less than one hundred millionths of an inch above the surface of the high-speed rotating disk by a cushion of air.

A machine error can be caused by a failure in the timing, speed, or movement of a transport mechanism or by a malfunction of the read/write unit. Such failures may result when devices get out of adjustment, are mishandled by operators, become worn, and so on. Failures may also be traced to bad media, such as warped cards, magnetic tape with surface defects, poor quality paper stock, or static electricity.

## **Equipment Controls**

Equipment controls are based on the concept of redundancy. In error control, redundancy involves the addition of circuitry, addition of an element to a process, or addition of an element to the code for an item for the sole purpose of detecting errors.

Equipment controls can be divided into five types: redundant character check, duplicate process check, echo check, validity check, and equipment check. Each of these controls involves a separate operation, which provides a check on the main operation.

### **REDUNDANT CHARACTER CHECK**

A redundant character check is a bit or set of bits attached to a data item for error detection and, in some cases, error corrections. A single bit is called a "parity bit," and a set of bits is called an "error correction code." The redundant character is developed from the characteristics of the data item to which it is attached. For example, after moving a data item from one location to another in the system, the computation used to obtain the redundant character is repeated to derive a second character. The two characters are compared; if they are the same, the transfer has not involved any malfunction. If there has been a malfunction, the redundant character may be used in an automatic error correction procedure.

### **DUPLICATE PROCESS CHECK**

In the duplicate process check, a process is performed twice, and the results of the two operations are compared. Any difference between the two operations indicates an error. The duplicate process may function as a complementary action, such as reading to check what has been written.

### **ECHO CHECK**

In an echo check, a device first receives a message or a command to perform an operation. It then returns a signal that verifies that the message has been received or that the mechanisms for performing the actions have been activated.

### **VALIDITY CHECK**

A validity check compares the result obtained of an operation with the set of valid

results. Any result not fitting into this set is incorrect. For example, a card reader may perform a character code validity check on each column of the input card.

### **EQUIPMENT CHECK**

In this control, the circuitry or equipment is checked to ensure that it is functioning properly. It is not a positive check, since the equipment may work properly while defective media and other factors cause incorrect results.

## **Central Processor and Storage Hardware Controls**

The central processor and the storage hardware have two problems. The first problem is to ensure that all data elements are transmitted through the internal circuitry of the central processor without loss or alteration. The second problem is to avoid the performance of an invalid operation. The basic central processor hardware controls are the parity check, automatic retry and diagnosis, validity check, and storage protection.

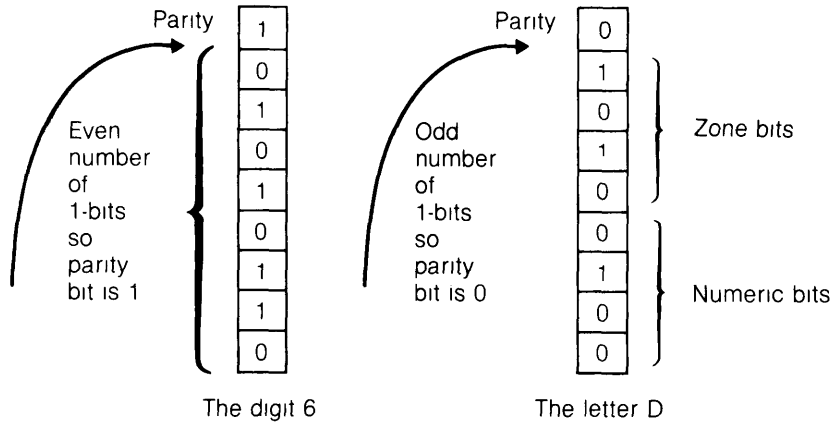
### **PARITY CHECK**

The computer stores, retrieves, and moves sets of binary digits, or bits, which are used to encode the data being processed. The design of the computer defines the number of bits in a basic set. A set of eight bits, called a "byte," is most common, but some computers use sixteen, thirty-two, thirty-six, and so on.

Bit values in storage may change due to some electronic malfunction or random disturbances affecting the storage medium. In addition, when the set of bits is moved through the circuitry, a bit value (1 or 0) may be altered by a circuit malfunction. The parity check requires that an extra bit, called a "parity check bit," be added to the basic bit set. When data are converted to binary form or when data are formed from a computation or other manipulation, the number of bits in each basic set is summed, and a 1-bit or 0-bit is placed in the parity bit position in order to make the total number of bits odd (for an odd-parity check). Figure 6-1 illustrates the parity bit added to a byte. Each time the basic group of bits is read from storage and moved through the circuitry, the parity is checked. If the newly computed parity bit is different from the parity bit stored in the parity bit position, an error has occurred. Of course, the test is not infallible, since the same alteration of two bits leaves the parity bit unchanged, but the probability of this happening is very low.

When a parity error is detected by a parity check, processing is interrupted unless automatic retry or error correction circuitry is available. In some hardware, parity errors set a hardware switch that must be tested by the program; if no test is made, the error correction procedures will not be performed. For a parity error, the normal procedure is to retry the processing because most parity errors are transient. If the same error occurs again, corrective maintenance is required.

In the case of parity errors associated with storage retrieval, many newer data processing computers add error correction codes, called "Hamming codes," to each unit of stored data. In a common approach to error correction at retrieval, the parity bits from eight bytes grouped together at storage form an error correction Hamming code for the group. When data items are retrieved, error checking circuitry analyzes the bits used to code the data. Single-bit errors are detected and automatically corrected. Multiple-bit errors are detected but not corrected.



**Note:** Bit coding for the characters is the American Standard Code for Information Interchange (ASCII)

**Figure 6-1** Odd parity coding for computer using eight-bit byte plus parity check bit

There are a few computers used for processing scientific data that do not employ built-in parity bit control. For some minicomputers, parity checking is an extra option or is not available. Some designers of minicomputers maintain that semiconductors or core memories are so reliable that the cost of parity checking cannot be justified. However, the major suppliers of data processing computers not only are continuing to use parity bits but are adding error correction Hamming codes to storage.

### AUTOMATIC RETRY AND DIAGNOSIS

A computer's electronic circuitry is subject to transient errors, which are errors caused by temporary conditions that may not recur, such as occasional variations in circuit switching times or static electricity. Since the processing will be performed properly after a retry, many computers have automatic retry facilities built into the hardware.

A fault may occur in one of the memory and circuit modules that plug into the central processor or storage unit. In such cases, the module is relatively simple to replace. Many computers are being provided with self-diagnosis capabilities to identify a defective circuit or memory module. The diagnostic facilities may be software only, but there is increasing use of diagnostic hardware as well.

A trend in medium- and large-scale computer design is to assign system management and error diagnosis to a separate support processor. (essentially a small computer.) The system management unit handles messages to and from the operator console, the loading of system software, and the partitioning of memory. In addition, the support processor performs error diagnosis and error handling.

### VALIDITY CHECK

In the central processor there are only certain valid operation codes that the computer can execute and only a certain range of numbers it can access as memory addresses.



Before attempting to execute an operation code or to access a memory location, the computer usually performs a validity check to determine that it is a valid code or address

### **STORAGE PROTECTION**

Most computers can execute two or more programs at the same time, which requires facilities for switching back and forth among the programs. This capability also requires some method of preventing one program from inadvertently overwriting the data or instructions belonging to another program. The methods currently used assign blocks of storage to programs. The assignment is checked by hardware and software procedures. If a program attempts an unassigned access, it will not be allowed, and an error will be indicated.

### **Card Reader Hardware Controls**

The card transport mechanism in a reader pulls a card from the input hopper and moves it past the read stations (usually comprised of photoelectric cells) at a precise speed and in a precise position. Therefore, a malfunction in the card reader may occur because the cards move past the reading stations at incorrect time intervals, because the cards are positioned incorrectly, or because the reading mechanism fails to sense properly. Card reading speeds vary from 200 to over 1,000 cards per minute, so that a slight delay or slight skewing can result in an incorrect sensing.

The control most often used in card readers is the dual read control, but a hole count control is also used. In the dual read control, the card is read by two separate read stations or is read twice by the same read station. The results of the two readings are compared. In the hole count control, the card is read by the first read station, and a count is made of the holes in each column or row, depending on how the card is read. A second hole count is made by a second read station, and the two hole counts are compared.

When the card reader detects an error, an internal switch is set. The typical error procedure is for the misread card to be routed to a separate output bin, called a "stacker." Depending on the application, the computer may either halt processing or continue processing with the error noted for later attention. If it halts, the reader will usually reject all cards then in motion. A corrected card and the other cards that were in process are then reinserted into the system. Since there is a greater chance of error when a card is reinserted into the system, many installations prefer to complete the run with errors noted for later attention.

### **Printer Controls**

The first step in printing a line is to assemble the characters in storage. These characters are then loaded into a print buffer, where they are decoded into signals that will select the characters to be printed at each print position. There are usually from 100 to 160 print positions on a line, with the most common range being between 120 and 132.

The two basic types of printers are impact and nonimpact. Impact printers employ a rotating drum or a horizontally moving chain, band, belt, or train that contains the characters. The characters are moved past the paper and ribbon that are in front of each possible print position. When the proper character is in position, a solenoid-driven hammer presses the paper and ribbon against the character. Matrix printers also employ impact, but characters are formed by the impact of a set of fine rods or wires selected from a matrix of wires.

## 88 Hardware Features for Control Over Equipment Malfunctions

High-speed, nonimpact printers generally use electrostatic printing methods, in which writing heads transfer "dots" of electrical charges to the paper. The paper passes through a toner, and colored particles adhere to the charged portions to form printed characters.

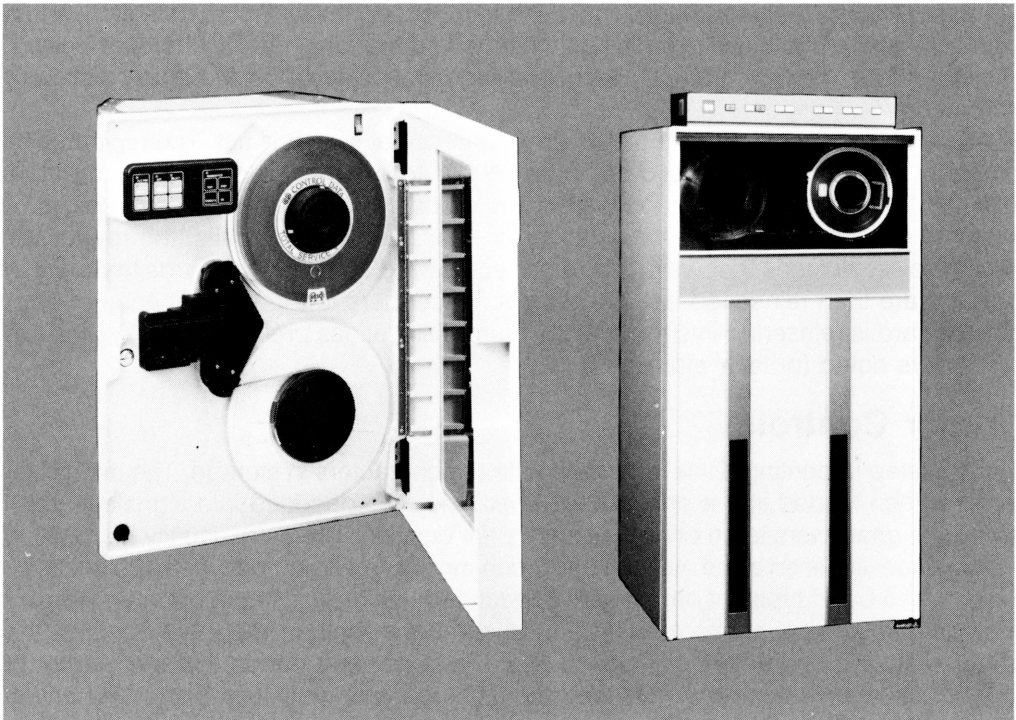
A printer may get out of adjustment and print wavy lines or partially formed characters. In some instances, it may fail to print certain characters altogether. A printer may also malfunction and print one or more erroneous characters because of timing errors.

Two hardware controls are generally used in printers. In the print buffer parity check, the character codes transmitted to the printer are tested for parity errors. The print synchronization control checks the timing of the printer to determine that the print hammers are activated when the appropriate characters are in the correct position.

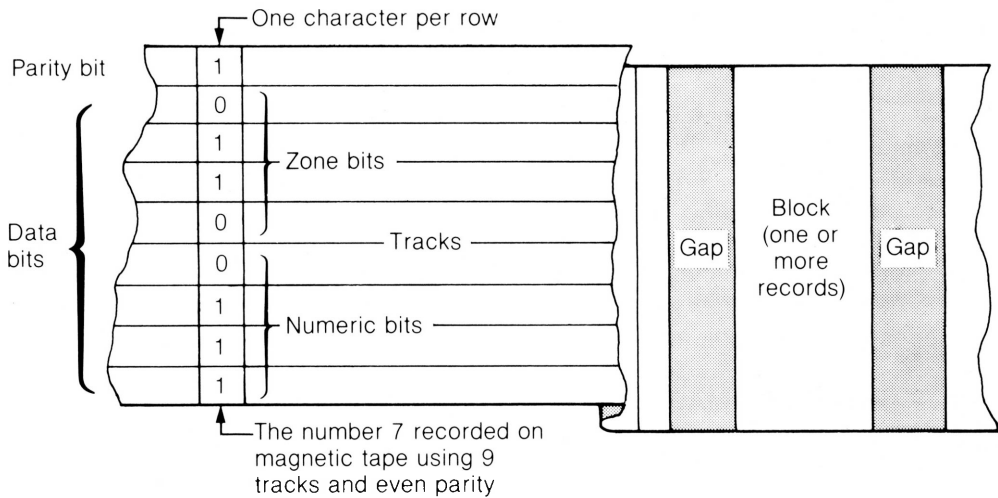
### Magnetic Tape Hardware Controls

A magnetic tape drive consists of a feed reel, a takeup reel, read/write heads, a pinch roller (or a vacuum or clutch-operated capstan) to drive the tape past the heads, and vacuum columns or mechanical tension arms to dampen the effect of rapid stopping and starting by providing several feet of slack tape for immediate movement before the reels start their independent motion. (See figure 6-2.) An alternative to tension-arm tape units is streaming tape, in which a short length of unsupported tape, controlled by servos, is moved on a cushion of air. It is a low-speed, low-cost device especially suited for backup storage. Tape may be on reels or cassettes.

Data are recorded by magnetizing small areas on the tape. A set of bits, encoding either one character or part of a computer word, is arranged as a vertical row of six or



**Figure 6-2** Vacuum-column tape unit (*right*) and streaming tape unit (*left*)  
*Courtesy of Control Data Corporation*



**Figure 6-3** Magnetic tape data record

eight bits on the tape. A parity bit position is added to each row, making the total number of bit positions equal to nine (figure 6-3). The horizontal recording positions are called "tracks."

Data records are stored on magnetic tape as physical blocks of coded characters, with each block containing one or more records. The data blocks are separated by gaps of .3 or .6 inches. The tape drive reads or writes a block each time the appropriate instruction is given. The gaps allow the tape drive to detect the beginning and end of the block, as well as allow it to accelerate for reading and to decelerate after reading. In addition to the gaps, there is a physical marker placed on the tape to identify the beginning and end of the tape for the tape drive. The small amount of space taken by the interblock gaps (when compared with the space taken by the data blocks) and the speed of acceleration and deceleration make large data blocks more efficient for tape storage.

Problems with magnetic tape may result from difficulties with the magnetic tape itself (the most common source of trouble), from malfunction of the read/write heads, or from malfunctions of the tape drive mechanism. The typical magnetic computer tape consists of a .5 inch-wide strip of plastic film 2,400 feet long with a thin iron oxide coating. As the tape passes over the read/write head, any imperfection in the coating may lift the head from the tape, causing a signal drop. This problem may result from a defect in the tape or from dust particles on the surface of the tape; however, the most frequent causes are the imbedded particles of oxide that flake off the coating of the tape and are redeposited on the tape surface by the tape heads and guides. If imperfections in the tape surface interfere with reading or writing, the tape is backspaced and reading or writing is retried. The retry may dislodge the particles and allow reading or writing to continue. These particles can also be removed by a special cleaning process, which should solve the problem altogether. Some tape units may have a tape scrape and tape cleaner that operates during rewind and unload operations.

In addition to the flaking of the oxide surface, there is another common problem in using magnetic tapes. Over a period of time, parts of a tape become worn. These sections must be either removed or marked so they will not be used for recording.

The proper operation of the read/write mechanism in the tape unit depends on the correct signals from the central processor and on the correct position and speed of the tape as it moves past the read/write heads. If the tape moves at an improper speed, the recording or reading mechanism will not operate properly. Faulty start/stop timing may cause an entire block to be skipped. A malfunction of the write head may cause a recording to be made at less than an acceptable signal level. If the drive mechanism that detects the markers at the beginning and end of the tape is not operating properly, the read/write heads may attempt to write or read on the leader at the beginning or end of the tape, resulting in incorrect processing or loss of data.

The hardware controls for magnetic tape consist of parity checking, error-correction codes, and tape unit monitoring.

**PARITY CHECK FOR MAGNETIC TAPE**

The basic parity check for magnetic tape is a row check, often called a "lateral check," in which each set of bits on a row is given a parity bit when the data item is put on the tape. When the data item is read from the tape, the parity bit is checked. Error correction is improved by the addition of longitudinal or track parity bits to create a two-dimensional check. Each block encoded on the tape is given a track parity bit in addition to the row parity bit associated with each row. The intersection of a missing row bit and a missing longitudinal bit will, if only one bit is in error, define the exact bit position causing the error and will allow for automatic error recovery (figure 6-4).

**ERROR CORRECTION CODES**

The low cost of magnetic tape and the high cost of tape errors justify error correction codes. Error correction codes are extensions of the Hamming codes used in the error correction codes for storage, and they are more powerful than the longitudinal parity bits. The error correction code is automatically generated during writing to tape and automatically checked (and errors immediately corrected automatically) during reading from tape. The error correction code is written after each unit chosen for error correction. The unit can be a block of data, but on high-speed, high-density tapes

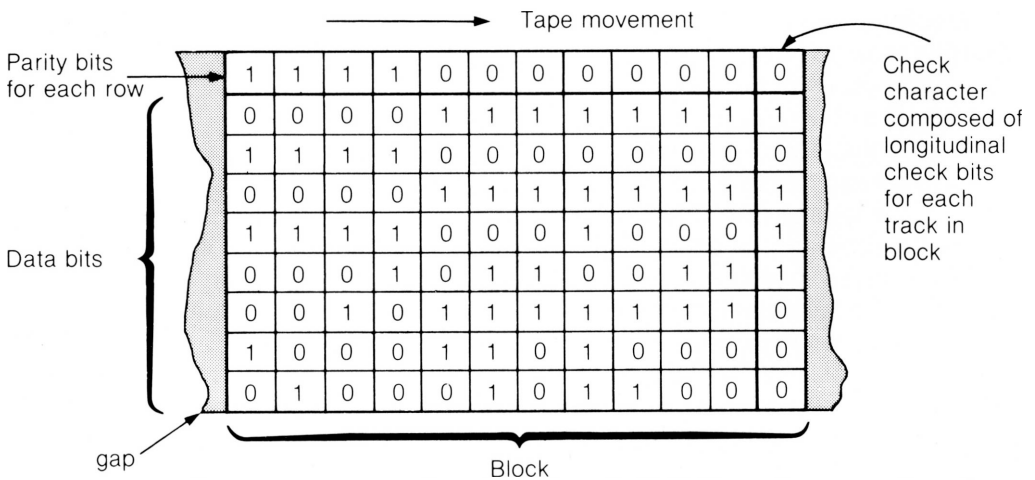


Figure 6-4 Magnetic tape parity check

(6,250 data bits per inch) there is an error correction code after each small group of data. In the latter cases, the ratio of error correction bits to data bits can be quite high.

### TAPE UNIT MONITORING

Some magnetic tape drives use a two-gap head to read the tape after it is written to check the recording. Current high-speed technology uses a single head to write and read. (Only one operation is performed at a time.) In order to check the tape while it is being made, the recording head is monitored to see that current is flowing through it. Other functions of the tape unit may be monitored by a built-in microprocessor that is programmed for checking and diagnosis.

## Disk Storage Hardware Controls

Disk storage consists of either rigid disks or flexible disks (diskettes). The flexible plastic diskettes are 8 inches, 5¼ inches, and 3½ inches and are used mainly with microcomputers. This discussion will focus on the higher-capacity rigid disks. Disk storage consists of a set of rotating metal disks with access arms that move read/write heads over the correct position on the disk. Data are stored on circular bands called "tracks" (figure 6-5). The access time before reading or writing can take place consists of two phases: (1) the seek time to position a head on the selected track and (2) the rotational delay while the desired position on the track rotates under the read/write head.

There are variations in disk storage design. The most common technology employs movable arms in the disk drive and removable disk packs. However, the following are some alternative designs:

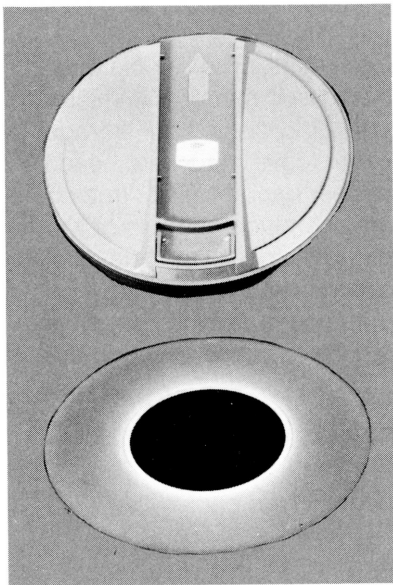
- A fixed read/write head over every track instead of an arm that moves in and out over the selected track
- A sealed disk pack with access arms and read/write heads in the pack rather than in the disk drive
- Fixed disks rather than removable disk packs

The tolerances in disk drives are very small. One common operating difficulty involves head crash, in which the disk read/write head makes physical contact with the surface of the disk. This can be caused by debris on the disk surface (generally dirt, grease, and oil particles circulated by the disk drive cooling fans), denting or warping of the disk (deviations in the axial plane of the disk), or damage to the rather vulnerable index (or sector) disk at the bottom of the pack. Another problem is head bounce, in which the disk read/write head does not maintain its proper position above the surface of the disk. Obstructions to the airflow between the head and the recording surface usually cause this trouble. A third problem, an inability-to-read signal, is caused by debris on the disk surface.

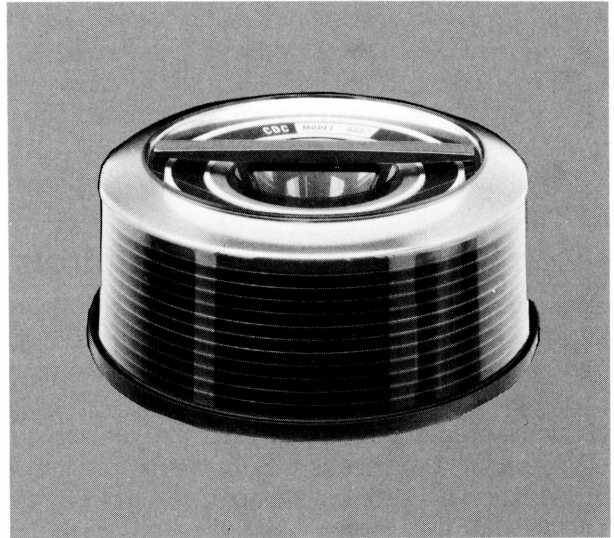
The major hardware controls for disk storage are parity check, error correction codes, automatic retry, and address checking.

### PARITY CHECK

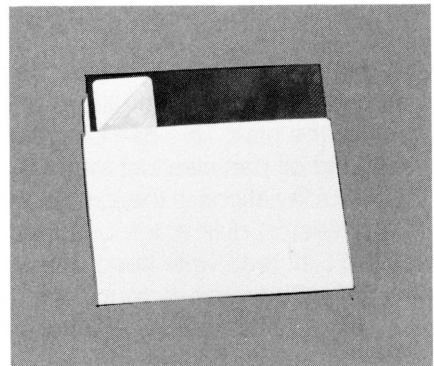
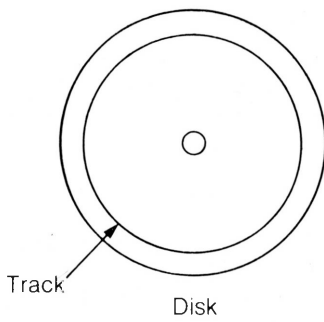
A parity bit is generated and recorded for each byte or other basic storage unit. When the data item is read, the parity is checked.



Disk cartridge



Disk pack



Floppy disk (in envelope)

**Figure 6-5** Magnetic disk storage  
*Courtesy of Control Data Corporation*

## ERROR CORRECTION CODES

A set of error correction bits is generated for a set of storage units. An error correction code is similar to the Hamming code or longitudinal parity check, and it allows the detection and the correction of multiple-bit errors. The use of the error correction codes can be quite powerful. For example, one manufacturer's error correction code allows automatic correction of up to nineteen consecutive bit errors.

## RETRY FACILITY

The hardware may provide automatic retry of a read or write. The unit will retry a read three to five times and then will automatically retry the read operation with different read/write head offsets. If the data bits were recorded slightly outside of the expected recording track (while the head was slightly offset), the retry with offset will be able to read them.

## ADDRESS CHECKING

When a disk pack is prepared for use, addresses are written on the disk as headers for all addressable units, such as track and sector. The address check is a hardware feature that compares the address in the address header with the address in the write instruction to detect incorrect disk read/write locations.

## Error Controls for Data Communications

The sending and receiving of data among devices at different locations is a common feature of computer data processing. The transmission of the data uses data communications facilities, usually provided by a data communications organization. Data are transmitted by an electrical signal or wave form that is defined by its strength (amplitude) and its duration (phase). The frequency of a wave form is the number of times the form is repeated during a specified interval. These three characteristics are diagrammed in figure 6-6. The signal is an analog (or electrical equivalent) of the communication input. If the basic shape of a wave form is known, a few measurements made at selected intervals (in other words, a sampling) provide the receiving instrument with sufficient data to identify the signals. In order to sample during the interval when a bit is transmitted, the receiving terminal must be synchronized with the sending terminal. The standard unit of signaling speed is called a "baud," which equals one pulse or code element per second.

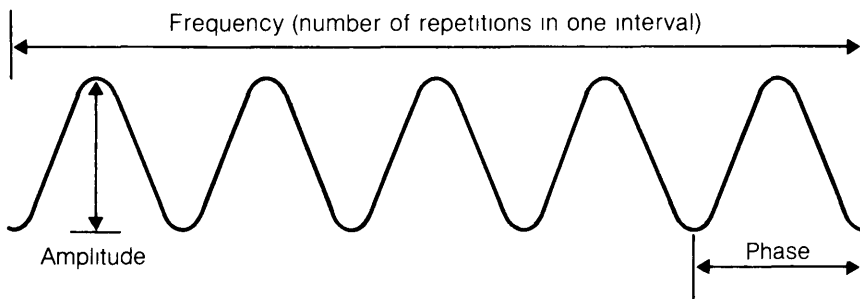


Figure 6-6 Waveform diagram

## 94 Hardware Features for Control Over Equipment Malfunctions

A transmission medium is usually separated into independent bands or data paths, each consisting of a range of frequencies assigned to it. The band width is the range of frequencies accommodated within a band on a transmission medium. For analog transmission, the constant-level, direct-current pulses of the computer equipment must be converted into signals suitable for transmission, and the reverse process must be performed for the receiving terminal. The conversion, called "modulation" and "demodulation," is performed by a modem or data set.

An electrical signal must be repeated or regenerated at intervals along the transmission medium to replenish its strength. The typical analog signal is amplified in whatever shape it is received, in other words, distortions are carried along the line. Another method of regeneration uses digital logic to analyze the received signal and to create a new, clean signal to send along the line. A communications network using this method is called a "digital transmission network." The method improves transmission speed and quality.

Communications companies offer either leased lines or public switched lines. (The term "lines" commonly refers to all types of communications facilities, even though they may use nonwire methods, such as microwave.) Leased lines provide users with a specific communications path for their sole use. The switched line (also called "dial-up"), on the other hand, provides access to the communications network. For the switched line, the path of connections and routing of a message may vary because automatic switching equipment in the network selects a path from the available connections and lines when the connection is dialed.

### CAUSES OF DATA COMMUNICATIONS ERRORS

Errors occur in data communications primarily because of inherent characteristics of the communication links or because of equipment malfunctions. Current data communication methods result in noise, fading, and distortion.

Noise refers to random fluctuations that interfere with the transmitted signal. The fluctuations may be steady background noise or impulse noise. Background noise, annoying in voice communications, is not usually a problem in data transmission. Impulse noise results from a disturbance (for example, lightning) that causes a burst of short-duration pulses.

Fading occurs in microwave transmission when signals are sent by sets of line-of-sight transmitters and receivers. Under certain atmospheric conditions, there is a fading in the signal received.

Distortion occurs because signals that travel different paths and in different parts of the frequency spectrum arrive at different times.

In general, a leased line has a lower rate of errors than a switched line because it is not subject to the variability of the switched connections. In addition, a leased line may be specially conditioned to permit higher data rates with a lower number of errors.

### METHODS OF ERROR DETECTION AND CORRECTION

There are a number of different methods for error detection and correction. The major methods are parity bit with each character, error correcting code for each block of data, retransmission, echo check, and diagnostics.

*Parity (simple)* A parity bit is added to each group of data bits (generally the bits to encode one character) to make the total number of bits odd in an odd-parity check or even in an even-parity check. Parity is checked when the group of bits is received.



*Error correction code.* One or more error detection/error correction codes are attached to each block of data. When the block of data has been received, it is checked. If there is an error, the error correction code allows for correction of either single- or multiple-bit errors, depending on the code used.

*Retransmission.* When a message segment (character, block, or message) has been received and checked for parity or other errors, the receiving station transmits an instruction to the sending station to continue with transmission if there is no error or to repeat the transmission if an error was detected.

*Echo check.* The echo check requires the transmission of each character in both directions. Data keyed from a terminal are not printed or displayed directly. Each character is transmitted to the computer, which sends it back to the terminal to activate the printing or display mechanism (termed "full duplex mode").

*Line and equipment diagnostics.* The latest modems and multiplexors contain diagnostic capabilities. One facility is a test pattern generator that provides comprehensive test input for the modem. Another facility is a loopback, which tests the terminal, modem, line, or remote device by looping back the output of the transmitting device to its input.

## Other Input/Output Devices

### MAGNETIC INK CHARACTER READER

Readers for magnetic ink character recognition (MICR), used on checks and other banking documents, have validity and timing checks built into the equipment. The timing check tests that the documents are moving past the scanning heads at the proper speed. As in other validity checks, the patterns read by the scanning heads are compared with the set of valid symbols. In addition, the circuitry may perform other checks. It may count the characters in the field to determine that the size of the field being read is correct or automatically compute a check digit and compare it with a check digit in the field being read. (The bank routing code now includes a check digit.)

### OPTICAL CHARACTER RECOGNITION EQUIPMENT

An optical scanner reads characters or marks with a beam of light. A character is identified by a particular pattern of light and dark areas. Two difficulties are often encountered in optical reading, one is measured by a reject rate, and the other by an error rate. The reject rate is the percentage of documents rejected because the equipment is unable to recognize the character. At present, reject rates range from one to two percent. The error rate is the percentage of documents that are read but contain one or more characters that were incorrectly identified. This error rate is very small.

The primary method for protection against incorrect identification is to add check digits to each field for which error detection is important, such as account number and amount. The check digit is applicable to documents prepared by the computer because the check digit can be computed and added to the line to be read.

All detected errors result in rejected documents. Automatic rescan and on-line correction are used to reduce the reject rate or reduce reject handling. Automatic rescan is used on page readers in which the document is held stationary. The scanner

## 96 Hardware Features for Control Over Equipment Malfunctions

reads the data three or four times before rejecting the document or skipping the characters as unreadable. For on-line correction, the unreadable character or the field with the unreadable character is displayed to an operator, who keys in the correct character or data item.

### **Power Supply**

The power supply for a computer sometimes fluctuates, which may cause errors in processing or shut down the equipment. This problem can be corrected by installing a motor alternator or an equivalent device, which makes corrections automatically when the power fluctuates.

There can also be interruptions of power supply. Standby power generating equipment may be used to protect against the possibility of interruption, but it is generally used only when uninterrupted processing is critical.

### **Hardware Controls and the Audit**

Even though the equipment is very reliable, hardware-based errors in data processing equipment do occur. However, there are satisfactory methods for detecting such errors and, in some cases, automatically correcting them. The auditor can usually rely on the equipment and the hardware controls for detecting errors. The auditor is interested, for background purposes, in identifying the major types of equipment controls and in checking that they are operating properly. This information is helpful for understanding the error procedures established by the client. The lack of a particular type of equipment control should not alter the scope of the audit unless there is evidence that the system does not operate within a tolerable rate of errors. However, the auditor should recognize the potential for errors involving magnetic storage media and take these into account when reviewing controls and designing tests.

A well run data processing installation is able to keep hardware-based errors to an acceptable level by such controls as failure logs and failure records, preventive and emergency maintenance, and error-control mechanisms in the equipment.

Data for equipment failure reports come from manual logs and from computer logs maintained by system management hardware and software. Regular review of equipment failure reports is important in identifying equipment in need of maintenance or replacement.

Regular preventive maintenance should prevent most equipment failures by identifying and replacing marginal circuit and storage modules. Both regular and emergency maintenance are improved by the use of diagnostic hardware and software in the equipment.

There is a trend toward automatic error detection and correction mechanisms, however, some hardware controls may require programmed tests to implement their use.

The auditor may obtain information on these procedures and controls by reviewing equipment failure reports and noting maintenance procedures. In most cases the hardware error detection methods are satisfactory and do not require special audit attention. Therefore, in reviewing processing procedures, the auditor should usually devote more attention to the procedures for handling errors than to the hardware controls that detect them.

## Summary

The auditor does not generally include hardware features in the scope of the audit unless there are indications that hardware-based errors are affecting the processing of data for the financial statements. However, the auditor should be familiar with hardware-error control and the potential for errors in magnetic storage media.



---

# System Software and Database Management Systems

# 7

In data processing, a distinction is usually made between application software and system software. Application software is specific to an application; system software performs tasks required by many applications. System software provides the environment in which application programs may operate. Two major categories of system software are operating systems and utilities, a third class of generalized software that is often described separately from others is the database management system.

## Operating Systems

The basic type of system software is the operating system (also called "executive," "supervisor," or "master control program"). An operating system is a complex set of program routines controlling the sharing of the central processing unit, storage, input/output devices, and software by application programs. Because it controls the operations that occur in the computer, the operating system can be used to meet many internal control objectives, however, its unauthorized use can circumvent many established controls, often without evidence that the circumvention has occurred. Therefore, a general understanding of the functions of an operating system and the methods by which the control features in the software meet internal control objectives can assist the auditor in the study and evaluation of internal accounting control.

### FUNCTIONS AND CONTROL FEATURES

The typical functions found in operating systems include the following

- Managing concurrent processing
- Scheduling processing
- Allocating resources
- Monitoring and controlling processing
- Handling errors
- Maintaining logs and computer accounting information
- Aiding data storage, maintenance, and access
- Aiding program storage, maintenance, and access

All of these features may not be available in every operating system. The functions and the way they are implemented vary widely with the software vendor and the size and age of the computer. Also, a user of an operating system may choose not to implement all of the available functions and controls. During installation (often called SYSGEN, or system generation), the operating system may be tailored to the needs and preferences of the individual installation. Often, the user may dispense with valuable control features for the sake of operating convenience or a reduction in processing overhead.

### **Managing Concurrent Processing**

A major function of an operating system is management of the concurrent execution of more than one program. For example, consider a system that is processing two programs at one time. The processor performs a calculation for program 1. When it needs data, program 1 is put in a "wait state" while the operating system arranges to get the data from the disk. While program 1 is waiting for data, the operating system assigns the processor to execute program 2, which performs a calculation. The completed calculation in program 2 is now ready for printing, so program 2 is put in a "wait state" while the operating system arranges for the printing and notifies program 1 that its data are available. The switching between programs takes place in a fraction of a second.

### **Scheduling Processing**

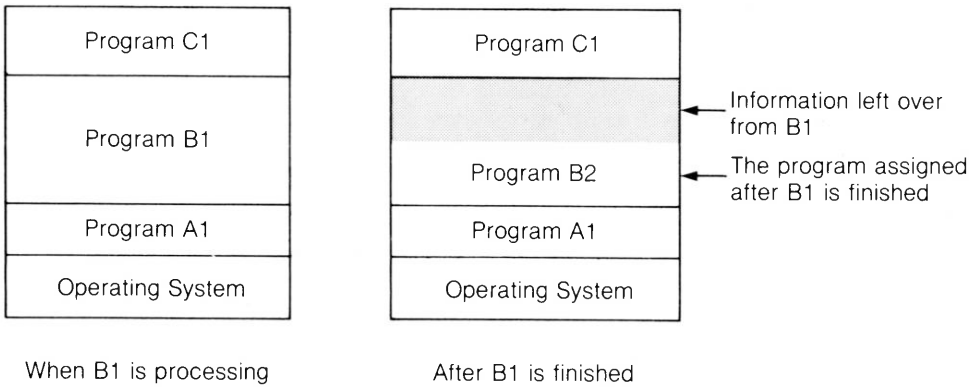
Scheduling in a batch environment is based on many different criteria. The system may take into account user-specified priorities, elapsed time, time of day, or conditions or events that have occurred. This function may be useful in controlling application processing. For example, the scheduling facilities of an operating system can be used to ensure that the program to print customer statements is not scheduled to be processed before the accounts receivable update program has been completed.

In an on-line system with immediate processing of transactions, scheduling is usually accomplished by assigning a priority to each type of event that may occur. The task with the highest priority is executed first.

### **Allocating Resources**

Two important resources allocated by the operating system are internal memory and input/output devices. There are various methods for assigning internal memory to a program. For example, some operating systems assign the program a fixed block of memory, the unused portion remains idle until the program completes processing. Other systems provide dynamic storage allocation by assigning only the amount of memory needed for each program. Storage protection features prevent a program from accessing memory that is not assigned to it.

A control problem may arise if storage is shared, because releasing a storage area does not erase the data stored there. Another program subsequently assigned to the area might be able to access residual data. For instance, as illustrated in figure 7-1, program B2 is able to read the information left in the shaded area from program B1. If program B1 is sensitive, the possibility of unauthorized access can be eliminated by including instructions in B1 to overwrite its area when it has completed processing or by having the operating system clear storage. This is important in systems where privacy is a significant concern.



**Figure 7-1** Internal memory allocation and problem of residual data

The methods used by different operating systems to allocate input/output devices range from simple to complex. Some systems assign devices to a program when processing begins and release them when processing is completed. Others assign the devices for only as long as the program needs them; during the time the devices are idle they can be used by other programs.

Other resources allocated by the operating system are data files and shared program files (for example, a subroutine that calculates square roots).

### Monitoring and Controlling Processing

The operating system initiates the processing of a program, synchronizes various processing activities, and, after processing is completed, closes files, reallocates resources, and records information for accounting and control purposes.

### Handling Errors

Many operating systems have the ability to check for errors caused by the hardware or software and to perform some corrective action. For example, if there is a data check in reading a magnetic tape, the system identifies the error and attempts to reread the record. If an error cannot be corrected by the operating system, it stores the information necessary to restart the processing of the current programs and stops all processing until the problem is resolved.

To handle software errors, the operating system either initiates an error handling routine designed to solve a minor problem or terminates the program to solve a major problem. The error handling facilities in the operating system are important controls, and they should be used to the fullest possible extent.

Some operating systems have testing and debugging aids (such as program tracing facilities and storage dump routines to display memory) to assist the programmer in handling errors. In some circumstances, these features may be useful to the auditor in performing audit procedures (chapter 14). However, the use of the testing aids should be controlled in the same manner as any other program to prevent them from being used to circumvent the processing control procedures.

### **Maintaining Logs and Computer Accounting Information**

Operating systems often record information to keep track of computer use and to charge users. They also gather statistics to analyze the operational efficiency of the data processing facility and system performance (chapter 15).

The computer accounting information provides a record of all system activity, and management review of this information can serve many control objectives. For example, management may analyze the log and selected data to—

- Verify that only authorized personnel have used particular programs, data files, or the computer facility itself.
- Verify that only authorized programs were run.
- Verify that production programs were run at the authorized time and the authorized number of times.
- Determine whether there has been unauthorized use, or attempted use, of remote terminals.
- Identify any efficiency or performance problems in the data processing equipment.

Use of the operating system accounting information feature may increase system overhead by from 2 to 5 percent, but the control it can provide is invaluable.

### **Aiding Data Storage, Maintenance, and Access**

The operating system performs four major functions in this area:

1. It provides facilities to set up data files, to allocate space for them on a storage device, and to locate a file when it is needed.
2. It provides methods to access stored data.
3. It processes internal file labels that have identifying information such as name, version number, date created, owner, and password.
4. It provides data file access controls whereby the owner of the file can specify who may use the file and how they may use it.

### **Aiding Program Storage, Maintenance, and Access**

Many operating systems have facilities to set up and maintain libraries of programs, subroutines, and sets of source code instructions, along with the library directories to locate them. In addition, the operating system makes the programs available for use and combines them as needed with the user's programs (sometimes referred to as "linkage editing"). For example, to code a program that needs to use a function included in the subroutine library, the programmer writes a statement such as "CALL CORRE," which calls the subroutine. At execution time, the operating system locates the subroutine CORRE, makes it available to the program, and performs all steps to link the subroutine to the calling program.

### **AUDIT CONSIDERATIONS FOR OPERATING SYSTEMS**

Because of the pervasive role of the operating systems in computer data processing, auditors may wish to consider them when reviewing internal accounting control. The review may be an examination of all general types of controls available in the operating system to establish which controls are in use. On the other hand, the auditor may choose to study the available controls through a review of operating system documentation, such as the vendor literature. This review can identify the control features that



are provided by the software and features that are useful for the auditor's purposes. The auditor can then determine which of the operating system controls have been implemented in the client's system, and he can review unused control features with management to determine whether there are compensating controls or there is control weakness.

An in-depth review of the operating system software and its coding is unwarranted unless the auditor suspects irregularities based on unauthorized system software changes and use. However, as explained in chapter 4, there should be segregation of functions between systems programming personnel who have detailed knowledge of the operating system and operating personnel who lack the knowledge necessary to make unauthorized changes.

## Utility Programs

Utilities are computer programs that perform common data processing tasks, many of which are termed "housekeeping tasks." Utilities are usually available from the computer manufacturer, but any program that is commonly used by the installation can be called a utility. The following are some common functions performed by utility programs:

- Copying files
- Changing the media of a file (for example, copying from disk to tape)
- Printing a complete file or selected items from a file
- Reorganizing the data on a file (for example, sorting or eliminating blank spaces)
- Setting up or purging files
- Generating test files
- Gathering statistics about usage of computer hardware, software, and data

### EXAMPLES OF UTILITIES USED FOR CONTROL

Many utilities perform tasks or generate reports that can be used to improve control. The following are examples of three areas in which utility programs are used to enhance control.

1. *Programs to improve documentation procedures*
  - Flowcharting software that reads program source code and produces a flowchart
  - Cross-reference systems that provide a list of all names used in a program and indicate the lines in the program where each name is referenced
  - Record layout software that converts program file descriptions to graphic or tabular format
2. *Programs to help provide adequate testing of new systems and changes*
  - Test data generators that create test records according to parameters specified by the user
  - Optimizers that monitor execution of a program and detect untested codes
  - Simulators that test a new system by performing the activities of an element (for instance, a terminal) that are not yet installed
3. *Programs to help management detect unauthorized usage or changes to programs and data*
  - Accounting data analysis facilities that capture data on the usage of computer resources in order to charge users

## 104 System Software and Database Management Systems

- Source code compares utilities that match two versions of a program and print a listing of changes
- Log analysis software that summarizes and analyzes history logs kept by the system

### **CONTROL CONCERNS FOR UTILITY PROGRAMS**

Utilities can be sources of unauthorized or undocumented changes to programs or files. Some utilities allow personnel to manipulate and change stored data without going through the control procedures established for application input and without going through the controls programmed in the application systems. Other utilities allow personnel to change programs without going through program change-approval procedures, testing, or other controls in the program change process. Editing and validation checks for data, password access controls, and program library controls can all be bypassed by using certain utility programs.

An example of a utility that can bypass controls is IBM's SUPERZAP. It can be used to make changes in machine language program code modules or turn off the password checking facility of the operating system. If SUPERZAP is not properly controlled, it can be used to make unauthorized modifications to programs or data with very little evidence of the changes.

Although utilities are useful, their uncontrolled use can threaten the security and integrity of data and programs. The potential risk in the use of each utility program should be evaluated, and those utilities that can be used to circumvent controls should be limited to selected personnel under controlled conditions. Some installations prohibit the use of high-risk programs like SUPERZAP or control their use by removing the utility from the normal system library and making it available only for authorized use. A record can then be kept showing the reason for its use and who authorized it.

Use of utility programs should be subject to the same controls as any other program. See chapter 8 for a discussion of the safeguarding of computer programs.

### **AUDIT CONSIDERATIONS FOR UTILITY PROGRAMS**

In reviewing the effectiveness of control over utility programs, the auditor can review the log of program usage (chapter 8). The auditor may also be interested in the use of some utilities in obtaining data for audit purposes and in assisting in other audit tasks. (See chapter 15.)

## **Database Management Systems**

Database management system (DBMS) software allows programmers to separate the way data items are used in processing (the logical structure) from the way data items are stored on the magnetic medium (the physical structure). The program requests data items as they logically relate to the processing being performed, and the database management system handles physical storage of the data and provides data to the program. The sequence and hierarchy of the physical storage format of the database on the disk or other magnetic medium may differ from the sequence and hierarchy of the data as described by the program.

## THE DATABASE CONCEPT

Database management systems are based on the concept that data records should be managed independently of programs that may create, access, or change them<sup>1</sup> The characteristics of a DBMS can be understood in terms of system objectives

- *Reduce data redundancy* When each computer application maintains separate files, a data item used by several applications will be repeated on several different files. A database management system, however, maintains a single file used by many applications. For example, using the separate data concept, the name and address of an AICPA member might occur on each of the following files:
  1. AICPA membership file
  2. Private companies practice section file
  3. *Journal of Accountancy* subscription file
  4. *Tax Advisor* subscription file
  5. *Practicing CPA* subscription file
  6. *CPA Letter* subscription file
  7. Accounts receivable file

Using the database concept, name and address are stored in a database, and the database management system maintains data elements for all member relationships.

- *Increase data consistency* When the same data items are stored in several places, the files are updated at different times and even by different sources. Sometimes the data may not be updated on all files. Because data redundancy is eliminated in a database system, only a single update is needed. In the above example, a single change of address updates the address for all member relationships.
- *Increase data independence from programs* Changing application programs to process new data elements is relatively easy in a DBMS environment because the data need not be stored physically in the sequence and format called for by the program. A change can be made to one program that processes the data without having to change all other programs that use the data. The data description for the program is changed rather than the physical storage of the data.

## THE COMPONENTS OF A DATABASE MANAGEMENT SYSTEM

A database management system may have the following components:

- Data dictionary/directory
- Data description language (DDL)
- Data manipulation language (DML)
- Query language
- Database management utilities

---

<sup>1</sup> DBMS software is sometimes used to maintain a separate database for each application system. This practice is an extension of the separate file concept, and the discussion in this section does not apply to such a system. The DBMS in that situation is merely another access method, and the system is not based on the database concept.

The data dictionary/directory is an optional software facility to organize and keep track of the information in the organization. It may be built into the DBMS or exist as a separate program. The data dictionary stores the definitions of both the computerized and noncomputerized data and keeps track of the location of each data element. By enforcing data definition standards, providing control over access to the database and the dictionary itself, and preparing data description lists and reports of data usage, the data dictionary can be an important element of internal accounting control.

The data description language is the language used to set up or change the database. It is used to describe to the database management system the physical characteristics of the data (alpha or numeric, floating point, length of data item, and so on) and the relationships between the data elements and the individual application programs. If a data dictionary/directory is used, its description of the data can be used to help develop the DDL specifications.

The data manipulation language defines the instructions used in the application program to call the database management system in order to access and manipulate the data in the database (for example, statements such as FIND, GET, MODIFY).

The query language is the general purpose language used to access the data for special, nonrecurring purposes. It is an especially important component for the auditor, who may wish to use it instead of generalized audit software. (See chapter 17.)

The database management utilities are the various software programs developed to assist in setting up and maintaining the database. The following are examples of database utilities:

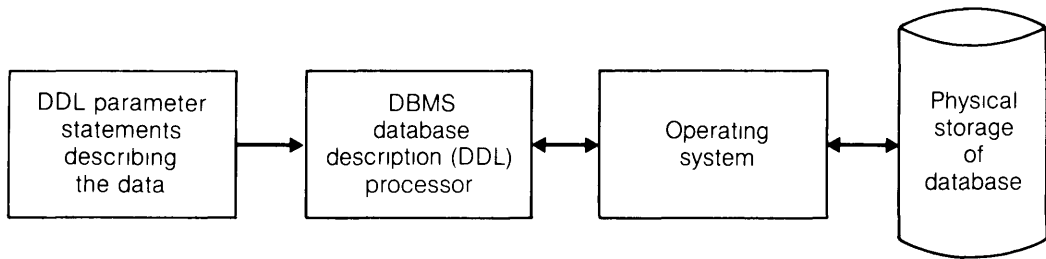
- Accounting and usage statistics, which record information to charge users and to improve operational efficiency
- Database dump/restore, which dumps or copies the program to another medium (usually tape) for backup and restores or reads the copy and recreates the database
- Database reorganization, which increases the efficiency of the DBMS by reorganizing the disk space to eliminate superseded records and by storing specific data elements (based on frequency of access and response requirements) where they can be accessed most efficiently

Database management utilities also include data-modifying routines (conceptually similar to SUPERZAP), which could be used to make unauthorized changes in data unless the utilities are controlled.

## HOW THE DBMS WORKS

Most organizations using a database system assign one or more database administrators responsibility for setting up and maintaining the system. To set up the database, the user departments, the application programmers, and the database administrator record the data to be included in the system in the data dictionary/directory. The database administrator incorporates this information into the design of the database and prepares DDL statements. The DBMS processes the DDL statements, and the database is set up and loaded (figure 7-2).

The application programs using the database have DML statements (such as GET or REPLACE) in the source code. When an application program is executed, the DML statements in the program pass control to the DBMS. The DBMS evaluates the parameters in the DML statements, performs some security checks, determines the location of



**Figure 7-2** Database setup

the data item in question, and passes on the appropriate instructions to the operating system. The operating system handles the activities and provides the data to the DBMS, which, in turn, supplies the information to the application program (figure 7-3).

### CONTROL ADVANTAGES OF A DATABASE MANAGEMENT SYSTEM

Certain aspects of internal control improve when a database management system replaces a conventional file system.

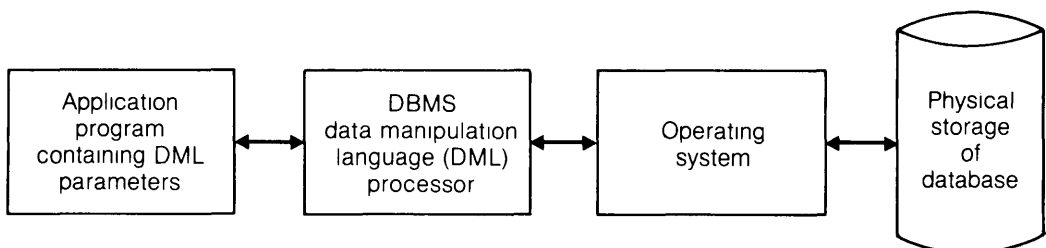
#### Data Consistency

One problem in using a conventional file system is to ensure that a data item is updated on all the files on which it occurs. By eliminating data redundancy, the DBMS ensures that all applications using the data item will be using the same value.

#### Timeliness and Availability of Data

The updating of data takes longer in a conventional file system than in a database system because of delays in sequencing and executing the file-to-file processing. The time lag can be reduced in a database system because data items are entered only once and are processed to completion at that time.

Data availability is also improved because the organizational data records are centralized, because the data items are organized, well defined, and clearly listed, and because the data items are independent of the application programs, so the data can be used in a variety of ways. Not only can the regular financial reports be prepared on a more timely basis, but special reports can be generated for planning and analytical reviews.



**Figure 7-3** Database use

### **Increased Data Integrity**

In a database system, data editing and validation are centralized. For example, instead of each application program checking that the account numbers are within the appropriate range and contain all numeric characters, the DBMS can have a standardized module to perform that function. Centralization has three advantages: (1) Standardized modules ensure that the editing and validation is consistently performed, (2) standardized modules are reliable and efficient, and (3) application programs are relatively easy to write because only those edit and validation tests that are unique for the application are coded.

### **Increased Data Security and Control Over Access to Data**

In setting up a database management system, a company is forced to consider what data elements are available and who should have access to them. The DBMS generally supplies standardized modules to control access to each data item and to check whether a user is authorized to change or delete the item. The recognition of the need for data security and the existence of facilities to perform the function increase the level of security.

### **Improved Documentation**

In conventional file systems, documentation can be scattered, inconsistent, or even nonexistent for some applications. In a database system, the data definitions and relationships must be clearly specified. Because data items are shared, all the programs using the data must have well-defined, documented data elements or the database will not function properly. The database administrator generally sets documentation standards and enforces them.

## **CONTROL CONCERNS IN A DATABASE MANAGEMENT SYSTEM**

Control concerns to be considered and compensated for in a database system are the concentration of data in one place, the single recording of a data item, the complexity of the storage structure and software, and the centralization of duties in the data administrator function.

### **Concentration of Data**

When most of a company's data records are gathered in one place, the risks of loss or misuse of data are magnified. If unauthorized access to the data occurs, all the records of the company may be violated. If a serious error occurs, all the company's data may be destroyed. Improved controls over access to data and backup and recovery facilities can compensate for the increased risks and may enhance control in these areas.

### **Single Recording of a Data Item**

Although the single recording of a data item for all applications provides consistency, it also multiplies the effect of any error that occurs. The single recording concept has several control aspects.

The responsibility for the accuracy of a specific data element may be diffused. Personnel in departments with little direct use of data may be assigned to provide input at a very high level of accuracy, yet have little motivation to ensure such accuracy. To

avoid such risks, management should consider (a) implementing the appropriate application controls described in chapters 10 and 11 and (b) establishing data ownership by assigning final responsibility for the accuracy of the data element to one department. In this way, the department that “owns” the data element has the responsibility to monitor the item carefully and to review the quality of the stored data periodically.

In a conventional file system, errors and irregularities are often discovered because of inconsistency of the data on one file with the data on another. The single recording of data reduces the opportunity to discover an error in this way.

When a data element can be accessed and changed by several application programs, the audit trail is much harder to follow.

### **Complexity of the Storage Structure and Software**

A higher level of technical knowledge is required to process, control, and audit database systems than to review applications with separate files. Many very different database management systems exist, and database technology is still developing.

Databases tend to be large. They grow and become more complex as management and other users discover the information resources available to them. Some databases have grown so large and unwieldy that they have been subdivided into several smaller, more manageable parts.

### **Centralization of Duties**

An important control technique is segregation of functions. However, in database systems, duties related to the data are centralized. The design and processing of the entire database is the responsibility of the database administrator.

## **THE DATABASE ADMINISTRATOR**

A company with a database usually has a database administrator (DBA) function with one or more individuals responsible for database administration. The more complex the computer system and data structure, the greater the need for a central function to coordinate the sharing of data among users and to oversee the database.

The database administrator function has the following responsibilities:

- To design the content and organization of the database, including logical data relationships, physical storage strategy, and access strategy
- To protect the database and its software, including control over access to and use of the data and DBMS and provisions for backup and recovery in the case of errors or destruction of the database
- To monitor the performance of the DBMS and to improve efficiency
- To communicate with the database users, arbitrate disputes over data ownership and usage, educate users about the DBMS, and consult users when problems arise
- To provide standards for data definition and usage and documentation of the database and its software

As these responsibilities indicate, the DBA is important for a well-controlled, smoothly functioning database system, but the concentration of technical knowledge

## 110 System Software and Database Management Systems

and operating responsibility provides the DBA with the opportunity to perpetrate a fraud or to conceal errors. Control of the DBA function can be improved by the following measures

### **Segregation of Duties Within the DBA Function**

Several people often perform the DBA function. Different tasks can be assigned to different individuals within the DBA group. For example, one person could be assigned responsibility for the logical structure of the data, another the physical storage structure, and another the access and search strategies. Because the various individuals use each other's work, this organizational structure helps to provide a check on each individual's activities.

### **Segregation From User Functions**

The DBA's duties should be segregated from all user transactions and applications. The DBA should not be able to initiate any transaction or to change a data element value (even for error correction) without the approval of the user responsible for that data element. The periodic review of each data element in the database by the appropriate data owner also helps to control the DBA function.

### **Segregation From Computer Operation Functions**

The DBA should not be allowed to run programs. When processing is required (for example, to change a structure in the database), the program should be submitted to computer operations, following appropriate and highly controlled procedures.

### **Segregation From Systems Development and Programming Functions**

When the DBA needs a program written, the specifications for the program should be submitted to the appropriate department for implementation. Ideally, the DBA should design the logical and physical data structure and access strategies, then give the design to systems programming for coding. The DBA would then review the results.

### **Maintenance of Logs and Reports of Database Activities**

A record should be kept (and reviewed regularly) of certain important database activities. For example, database utilities can be used to circumvent controls and to change or delete data. The use of these programs should be recorded in the log of program use. (See chapter 8.) Data processing management should review the log for any unusual use of the database utilities. Similarly, the data processing manager as well as the DBA should review appropriate reports generated by the data dictionary system, such as the report showing which programs used each data element.

### **Monitoring of Data Administration Activities**

A control function is used to monitor various data processing activities. The control function may be responsible for the following checks of data administration.

- Review and approval of the DBA's assignment of responsibility for data content to a user department and assignment of data element access rights
- Review of reports and logs of database activities



- Review of the DBA standards for data definition, documentation, and so on, and the extent to which they are being followed

### Care in the Selection of DBA Personnel

The person responsible for the DBA function must have technical expertise and skill in management and human relations. To maintain adequate control, the DBA should demonstrate integrity and be bonded.

## AUDIT CONSIDERATIONS FOR DATABASE MANAGEMENT SYSTEMS

The use of a database management system affects both the auditor's study and evaluation of internal accounting control and substantive tests.

### DBMS Accounting Control

Segregation of data administration functions

User data controls

Accuracy controls

Error correction controls

Access controls

### Audit Consideration

- Segregation of data administrator from incompatible functions
- Segregation of duties within data administration, if possible
- User responsibility for data
- User review of all changes
- Periodic review and comparison with physical counts and other evidence of correct data values
- Use of standard data editing and validation procedures
- Use of procedures for error correction
- Use of procedures to limit access to authorized programs to authorized personnel only

Because of the concentration of records in the database, the auditor and management may also wish to consider including in the audit scope a review of whether the DBMS backup and recovery facilities are adequate and functioning properly.

The use of a DBMS affects the auditor's ability to access the data for tests of transactions and balances (Chapter 17 discusses audit techniques for database management systems.)

## Summary

Generalized data processing software can be used to enhance internal control. Operating systems have facilities for conditional scheduling, error handling, recording of usage and computer accounting information, processing of internal file labels and passwords, and control of access to programs and data. Certain utility programs provide documentation procedures, facilities for testing of new systems and changes, and facilities to detect unauthorized usage or changes to programs and data. Database

## 112 System Software and Database Management Systems

management systems can enhance control by improving the consistency, timeliness, availability, integrity, and documentation of data

Operating systems, utility programs, and database management systems, however, all have features that may be control concerns. Therefore, the use of generalized data processing software should be controlled.

---

# Safeguarding the Availability, Access, and Use of Computer Facilities, Programs, and Data

# 8

The provisions that safeguard the availability, access, and use of computer facilities, programs, and data protect against (a) unauthorized access and use affecting current data processing, which would affect the financial statements, and (b) loss or destruction affecting future availability of data processing capabilities. The protection of current processing is included in accounting control and is within the required scope of the auditor's study and evaluation of internal accounting control. As stated in the audit guide, *The Auditor's Study and Evaluation of Internal Control in EDP Systems*, the general access controls are as follows:

- Access to data files and programs should be limited to those individuals authorized to process or maintain particular systems (general control 14)
- Access to computer hardware should be limited to authorized individuals (general control 15)

Threats that can result in destruction of computer facilities, programs, and data include disasters, equipment malfunctions, and human errors or abuses. Disasters are disturbances in the environment—natural, political, and technological. They include fires, floods, civil disorders, and power failures. Equipment malfunctions generally halt processing without damage to programs or data, but they sometimes can be destructive. For example, a disk device that contains an undetected head crash may write unreadable data. In another instance, a magnetic tape drive can, through a slight timing error, begin to rewind before all the slack has been taken up, which would result in a backlash that may peel off some of the oxide coating, stretch the tape, or break it. Human errors and abuses can also be destructive. For example, the careless mounting of an incorrect file, the dropping of a disk pack, the theft of machine-readable files, or malicious destruction of equipment can all result in the loss of data.

The controls to protect against loss of data or to ensure recovery of processing capabilities can be divided into the three following categories:

- 1 Safeguarding of the computer facilities, including physical security of the equipment and files and controls over access to the computer room
- 2 Safeguarding of the computer programs and data, including controls over accessing, using, and altering programs and data
- 3 Assurance of continued operations in the event of damage or disaster, including backup and recovery procedures for facilities, programs, and data

## **Safeguarding Computer Facilities**

A basic consideration in ensuring adequate processing of the computerized portion of the accounting system is the continued availability of the computer equipment and files. The physical safeguards include the following:

- Location, layout, and construction of the computer room
- Fire prevention, detection, and fire-fighting techniques
- Insurance
- Housekeeping and preventive maintenance
- Physical safeguards for files
- Control of access to the computer room

### **LOCATION AND CONSTRUCTION OF THE COMPUTER ROOM**

The location, layout, and physical construction of the data processing department can substantially affect the risk of damage. For example, locating the facility at street level with large plate glass windows in full view of passersby, placing the computer room in a building on a flood plain, or installing the facility next door to the kitchen of a restaurant would increase the risk of damage to computer data and equipment.

Some large corporations consider the likelihood of earthquakes, floods, and other natural disasters in determining the geographic location of their major EDP facilities. The risks of physical damage can also be reduced by selecting a secure site within the organization's offices. The space next to or above or below a kitchen or boiler room or an area in the basement or on the top floor of a building presents the greatest risk of damage. The center of the building, away from outside walls, is the most desirable location for a computer room. The walls should be fire and water resistant, and the floor should allow for drainage (of water used to extinguish a fire).

### **FIRE PREVENTION AND DETECTION AND FIRE-FIGHTING TECHNIQUES**

Tape files, card files, and disk packs can easily be destroyed by fire. These files are more susceptible to fire damage than the printed or written records of manual or tabulating systems because a small fire that only chars the edges of a stack of paper or pages of a book can melt a tape or warp a disk. In computer equipment, overheating to as little as 140° F can cause a malfunction of some transistors, and temperatures above 300° F can result in permanent damage to the system. The auditor should not assume that the standard fire detectors used in an office building will protect the computer facility from fire. Many office detectors are set off at 190° F, and at that temperature damage will already have occurred by the time an alarm is given. If management has not checked the type of fire detectors installed, the auditor may wish to advise them to do so. Fire detectors should also be placed to detect fires in spaces above false ceilings and below raised floors.

The National Fire Protection Association has made extensive recommendations concerning computer installations.<sup>1</sup> In general, these call for the following:

---

<sup>1</sup> National Fire Protection Association. Standard for Protection of Electronic Computer Systems. Bulletin No. 75 (Boston: NFPA, 1976).

- Housing of the computer in a noncombustible environment
- Use of smoke or heat detectors
- Availability of extinguishers
- Storage of vital records in cabinets having a class C rating (one hour at 1700° F)
- A separate air-conditioning system for the computer or a shut-off switch for cutting off the air-conditioning fans
- A separate emergency switch to control electrical power for the computer system
- Personnel trained in fire control procedures

Fire extinguishing systems for computer facilities may use water or carbon dioxide, however, there is a preference for the use of special gasses, such as Halon. While carbon dioxide is an effective extinguishing agent, it can suffocate the computer personnel. Although water may not damage tapes, disks, and computer-output microfilm, it may damage the computer equipment to some extent.

Programs and documentation, as well as data, should be included in the provisions for fire protection. In one case, a company purchased an excellent vault for storing magnetic tape, which they felt would provide adequate protection against fire. However, the company's programs on disk and backup tape were stored in a steel cabinet next to the computer. In the case of a fire, the company's data files would be reasonably secure, but the programs needed to process the files would probably be destroyed.

## INSURANCE

Insurance should be included in the plan to safeguard facilities and files. Adequate coverage can be developed by a competent insurance agent, but it is a good idea for an auditor to be aware of the types of coverage available so that data processing insurance can be included in the audit review.

The risks against which the organization should be protected arise primarily from fire and, if work is performed for others, from liability for errors or omissions. The risks from fire are covered in varying degrees by the three different types of policies: fire insurance, valuable-papers and records insurance, and data processing insurance (all risk) with a media and records section. A summary of coverage by specific risk is summarized in table 8-1. This table shows that data processing risks from fire are covered by regular fire or valuable-papers insurance, whereas the all risk processing insurance is designed specifically for the losses that are associated with computer data processing. The additional coverage is available through special data processing policies, special endorsements of regular policies, or difference-in-condition (DIC) policies.

Organizations (service centers, banks, CPA firms) that rent time or provide data processing services for others incur liability risks from (a) losses caused by errors in the work performed and (b) losses caused by programs written as a consultant. Both of these risks can be covered by data processing liability insurance.

Because of the concentration of duties in the EDP function, bonding of data processing personnel is a sound management practice. It provides security checks on new employees performed by the bonding company (if specifically requested) and compensation for a loss (if it occurs).

## HOUSEKEEPING AND PREVENTIVE MAINTENANCE

A clean, well-organized computer room reduces the threat of fire and the possibility of using incorrect files. During handling, the files should be identified and placed in speci-

Type of Risk	Fire Coverage	Valuable Papers Coverage	Data Processing Coverage
Damage and/or loss of equipment whether leased or owned	Cost of equipment	None	Cost of equipment
Loss or destruction of programs (software)	Cost of materials (cards, tapes, disks) and labor (key-entry) No coverage for costs of programming or systems design	Extent of coverage in doubt May exclude recovery for loss of data or tapes	Cost of reconstruction under critical conditions, provided that remote storage is employed for key files and documentation
Loss or destruction when reconstruction will be costly, time consuming, and difficult	Cost of materials on which data were recorded	Extent of coverage in doubt	Cost of reconstruction under critical conditions provided that remote storage is used
Extraordinary expenses incurred to return to normal operation	None	Extent of coverage in doubt	Covered
Loss sustained by interruption of business	None	None	Covered

**Table 8-1** Coverage provided by different policies for data processing fire risks

fied locations. For example, the operator should have different locations for files to be processed, scratch files, output files, and files that have read/write errors or that have been set aside to be cleaned. Good housekeeping procedures can help prevent a master file from being mistaken for a scratch file to be written on.

Preventive maintenance is also an important part of safeguarding the computer facilities. It should be performed regularly by qualified service personnel and should include the following:

- Detection of potential problems and appropriate adjustments
- Repairs as needed
- Maintenance of an inventory of spare parts, or knowledge of current suppliers
- Cleaning of read/write heads

### PHYSICAL SAFEGUARDS FOR FILES

Physical file protection depends on heat and humidity control and safeguards in data handling.

Magnetic tapes and disks have low tolerances for heat and humidity. An improper environment can stretch or warp a disk so that the read/write head of a disk drive will not be in the proper position to access the data it is seeking. Tape may become brittle if the room is too dry or is subjected to rapid changes in temperature. If the humidity is

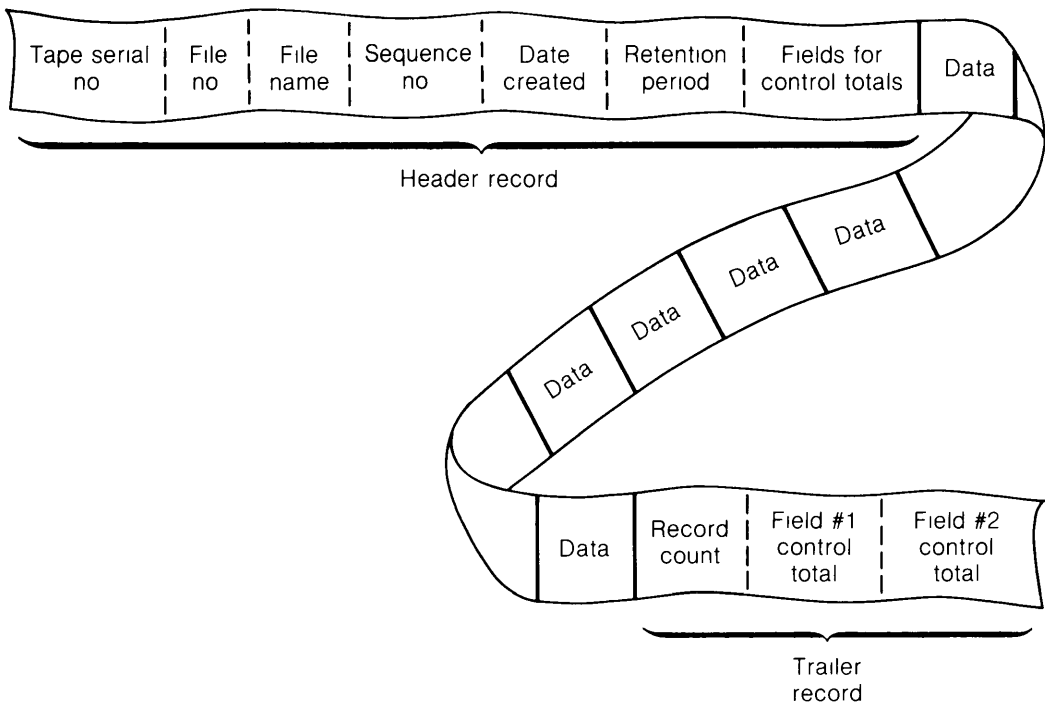
too high, paper products, such as punched cards and printer paper, expand and may jam in the card reader or printer. Regulation of the temperature and humidity in the computer room and use of vaults or safes that maintain a low temperature in a fire (as defined by appropriate Underwriters' Laboratories ratings) can help avoid these problems.

Safeguards in handling files include the use of internal and external labels, file protection rings on tapes, and read-only switches on certain disk drives.

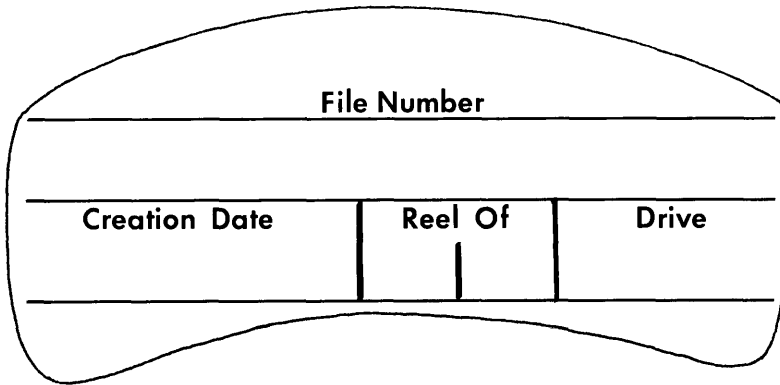
An internal label is a record containing identification and control information and can be used with both tape and disk files. On tape, a header label is recorded at the beginning of a file to provide identification, retention release date, and so on. The label plus checking procedures can protect files from inadvertent modification or destruction. A COBOL file label is an example of a header label. A trailer label is written after the last data record and contains record counts and control totals for the file.

Figure 8-1 illustrates the form for header and trailer labels on magnetic tapes. The standard procedure in most operating systems is to write a header label on all output files before writing data records and to read and check the header labels on all input files before reading data records. The COBOL file instruction OPEN, when used with the clause LABEL RECORD IS STANDARD (or standard label as default option), causes a label record to be written on an output file and to be read from an input file. Trailer label records are generally written and read by the application program. They provide control totals for procedures to detect errors or omissions in reading records from a file.

External labels help to ensure that the operator uses the proper file. They stick on the outside casing of files, indicating such identifying information as the file name and



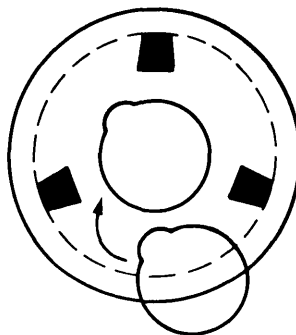
**Figure 8-1** Internal magnetic tape labels—header and trailer records



**Figure 8-2** External magnetic tape label

the date created. An example of an external tape label is shown in figure 8-2. It is also possible to use the color of the plastic tape reel itself as a further identification, and reels can be obtained in many different colors. An installation may work out an identification code, such as red for master files, yellow for system files, and so on. However, this safeguard is not a substitute for external labels. If the tape management procedures include strict internal label checking procedures, descriptive external labels are not necessary. Only the volume number of the reel is required.

Another physical safeguard used to prevent erasure of information on magnetic tape before the release date is the placement of a removable plastic or metal ring on the back side of the tape reel. The rings generally are inserted into a recess in the tape reel to allow writing and are removed from the reel to inhibit writing. When the reel with a write-enable ring inserted into a recess is mounted on a tape drive, it pushes a switch that allows the drive to write on the tape (figure 8-3).



**Figure 8-3** File protection ring for magnetic tape (write enable)

A similar safeguard also exists for some disk drives. They have a read-only switch that the operator turns to the read-only position when mounting a disk that should not be modified. Unfortunately, there are usually many files on a disk and this feature is practical only if all the files are read-only.

On microcomputers using diskettes, a diskette can be made "read only" by covering (or uncovering) a small notch with an adhesive strip.



## CONTROL OF ACCESS TO THE COMPUTER ROOM

To help avoid loss from human errors and abuses, access to the computer room should be limited to the authorized computer operations personnel. Chapter 4 stresses the importance of the segregation of functions between the programmers and systems analysts and the computer operators. This segregation must be implemented by restricted access to the computer room, or the control will not be effective.

Restricted access not only will provide better segregation of functions but also will improve controls over system development and the use and alteration of programs. Programmers operating the computer often cause disruption in the processing schedule and may disturb or circumvent the operating policies and procedures.

Maintenance personnel from the computer vendor, independent and internal auditors, and other individuals may require access to the computer room. Each visit should be authorized by data processing management and recorded in a log, and the visitor should be accompanied by an operator while in the computer room. (See table 8-2.)

The door to the computer room should be locked. Key locks are a common method to restrict access to authorized personnel, but many other techniques are used. Authorization can be verified in the following ways:

- Badges for authorized personnel and for visitors, which are checked by security personnel
- Magnetic stripe cards that allow a person with an electronically encoded card to enter the computer room (For some units, the individual card can be recognized and logged for review by management.)
- Use of a password or personal identification number checked by software (This technique is more common in controlling access to a terminal than in controlling physical access to the computer room.)
- Combination locks of varying degrees of sophistication
- Use of fingerprints or handprints (In a typical device of this type, the person seeking access keys in a number or inserts an ID card to indicate to the system the set of personal characteristics it will check. The user then puts his hand on the reader,

Personnel	No Access	Access When Authorized and Accompanied by an Operator	Free Access During Appropriate Hours
Users	X		
Data preparation personnel	X		
I/O control clerks	X		
Librarian	X		
Application programmers	X		
Systems programmers		X	
Vendor's maintenance personnel		X	
Auditors		X	
Outsiders renting computer time		X	
Operators			X
Cleaning personnel			X

**Table 8-2** Computer room access

## 120 Safeguarding Availability, Access, and Use of Computer Facilities

which scans it and checks the results against the characteristics on file for that person )

- Signature verification systems (Rather than matching the appearance of the signatures, these devices measure the amount of pressure applied to the pen )
- Voice verification systems (A voice recognition unit compares voice patterns with access words to a stored pattern for each authorized individual )

These techniques can also be used in combination. For example, some devices that use magnetic stripe cards also require the person to enter an identification number. This helps to guard against security risks from lost cards.

Access to terminals can be controlled by both physical access procedures and program/data access controls. A terminal performing sensitive functions, such as large funds transfers, may be kept in a controlled access room. In such cases, physical control is used along with programmed access controls. If a terminal is accessible to unauthorized users and programmed identification methods are insufficient, a key or card may be added for more access control to unlock the terminal.

## Safeguarding Computer Programs and Data

The controls to protect the physical facilities and terminals are only one part of the controls to safeguard processing. The programs and data stored in the computer and on the files should also be protected from unauthorized use and modification. Safeguards for programs and data include the following:

Librarian function

Passwords and other identification techniques

Encryption

Well-controlled program change process

Operator procedures to control the use of programs and data

Library software and other security software

### LIBRARIAN FUNCTION

Management can control access to documentation and data that are not in use by assigning responsibility for access to an individual or group. This librarian stores programs, data, and documentation that are not in use in a physically secure area. Access to this area should be restricted to the librarian. The librarian releases files to authorized personnel according to established procedures or on receipt of approved written requests; for example, he may release the payroll files to the operator every Thursday at 9:00 A.M. He logs the issuance of data and documentation and the receipt of the material when it is returned (figure 8-4), and he logs the use of each physical tape or disk to ensure the quality of the recording media (figure 8-5). For each tape and disk, the librarian should record each use, any errors that occurred, and each cleaning. The librarian should also take periodic inventory of program files, data files, and documentation. Finally, the librarian maintains appropriate backup copies.

All EDP facilities should assign a librarian function. Large organizations assign several people to the function full time, either separately or as part of a control group. In a small data processing department, the function can be assigned as part of the duties

Date	Time Out	Time Returned	Released to (Name)	File No	Tape or Disk Serial No	File Name	Reason Scheduled (S) or Approved Request (A)	Errors	Comments
1 '24	9 30		Felipe	'3798	6242	INV99 Inventory (Test)	A		
1 '24	9 35	11 40	Clarke	'6352	'864	PAYROLL Master	S		
•	•	•	•	•	•	•	•		
•	•	•	•	•	•	•	•		
•	•	•	•	•	•	•	•		

**Figure 8-4** Librarian's log of program and file use

of one or more individuals<sup>2</sup> In the hours when the librarian is not available, the files scheduled for processing could be left with the shift supervisor, and the remainder of the files could be locked in a secure area. An emergency procedure may be necessary to access files when the librarian is not on duty, but should be designed to leave evidence that access has occurred. Follow-up procedures can then be performed (for example, review of the job accounting data) to detect any unauthorized processing occurring during that shift.

## PASSWORDS AND OTHER IDENTIFICATION TECHNIQUES

Access to on-line programs and data can be controlled by the use of identification techniques, such as passwords. Usually, passwords are stored on an internal table accessible only to the operating system. The user includes the password in his sign-on procedures or in the job control statements to obtain and run the program.

Some password security systems require different passwords for different classifications of data and programs or for different functions to be performed with the accessed file. For example, passwords can be required to perform the following tasks:

- Access the system
- Access high-, medium-, and low-security files or data elements
- Read programs or data
- Read and modify programs or data
- Add new programs or data files
- Delete programs
- Transfer files from one classification to another

Passwords are effective for access authorization and access security only if certain conditions are met. The password must be kept secret, so that it will not be learned by unauthorized personnel, a password should not be written down and should not be given to another user. Passwords must be difficult to guess, names, birthdays, and so on should not be used. Rather, the system should generate passwords composed of random characters. Passwords should be changed periodically, whenever a person knowing the password changes position, and whenever there is suspicion of security violation.

<sup>2</sup> In this case, the auditor should consider whether the other duties of the part-time librarian are compatible with this function.

Tape History Card					
Date purchased		Mfg	Length	Reel number	
Density		Date	Length stripped	Remaining length	
Date certified					
Date out	Program no	File no	Program and file name	Errors	

Figure 8-5 Tape reel history card

A password security facility should monitor all unsuccessful attempts to use protected files<sup>3</sup> Repeated attempts to access a file with incorrect passwords could indicate that someone is trying to guess the password or to exhaust all the combinations of characters until one works Logs of the attempts should record date, time, location of terminal (if applicable), and the information entered in the attempt Some password security procedures can also shut down any terminal that logs repeated attempts to use incorrect passwords or to use certain instructions A terminal shut down for this reason should be required to be re-established by a supervisor

## ENCRYPTION

Encryption safeguards information by encoding it Even if an unauthorized individual were to acquire the data, it would be meaningless

Encryption is accomplished either by a secret encoding process or by using a standard algorithm (the set of steps to perform the encryption) with a secret key Most of the encryption systems use the latter method The user enters the data in its ordinary form, along with a secret parameter or key to be used in the algorithm The algorithm may scramble the order of the characters, substitute characters for other characters, compress the message, or add meaningless characters for confusion The information can be translated to meaningful data only by a decoding process using the exact key used to encipher it A different key results in a different interpretation of the information, which will be meaningless

The National Bureau of Standards has published a standard algorithm for data encryption<sup>4</sup> A facility should consider hardware devices rather than software to implement the standard algorithm, a distinction that will—

- Increase efficiency The elapsed time and resources used are many times greater if software is used
- Reduce the possibility of disclosure of the key The key can be stored within the device rather than on a file that is accessed by the computer
- Facilitate testing and monitoring because the key is isolated from the computer software system
- Reduce the possibility of unauthorized modifications Modification of software is easier to accomplish and harder to detect than modification of hardware

The cost of an encryption system increases with its effectiveness The decision to use encryption techniques, and the choice of the specific system, depends on the degree of security needed for the data and the potential losses if the data records are disclosed One common use of encryption is to protect sensitive data being transmitted through a data communication system

## PROGRAM CHANGE PROCESS

Management should establish policies and procedures to control modifications of computer programs<sup>5</sup> Unauthorized modification of programs, whether accidental or inten-

---

<sup>3</sup> Generally, the facility is part of the operating system, communications system, or library software

<sup>4</sup> National Bureau of Standards, Federal Information Processing Standards Publication 46 *Data Encryption Standard* (1977)

<sup>5</sup> See Computer Services Guideline, *Control Over Using and Changing Computer Programs* (New York: AICPA, 1979)

## 124 Safeguarding Availability, Access, and Use of Computer Facilities

tional, can render them useless, cause incorrect processing, and permit errors or irregularities

Program changes should be subject to all the control procedures used in systems development (See chapters 4 and 5 ) The systems and programming standards manual should include a section on the procedures to implement changes, and those procedures should be enforced This means that programmers should be forbidden to patch programs that do not run to completion In particular, all programming changes should be approved, documented, and tested before they are used on live data

To achieve these control objectives, many companies have established two separate libraries or classifications of programs production programs and test programs Production programs are the authorized versions of programs used in the regular processing of data They are segregated from the test programs, which are often renamed copies of the production programs upon which changes are made and tested No test program should be given production status until testing has been completed and management has given written approval The operator, rather than the programmer, should place the program in the production libraries, and programmers should be prevented by controls within the software and by manual procedures from accessing the production libraries The program it replaces should be copied to a backup file and then deleted from the production library All inactive programs should be deleted and archived to reduce the possibility of processing the wrong program

These techniques are also used to help prevent unauthorized changes to programs Controls should also exist to detect such changes if the other controls fail. On a test basis, data processing management can review the console log or job accounting data for unusual activity Library software can also be used to generate reports to help monitor the program change process

Another method to help detect unauthorized changes to programs is a hash total of the program code The hash total is the sum of the numeric representation of the characters in each line of the program When an authorized change is made, a new hash total is computed Periodically (or when suspicion of an unauthorized change arises) the hash total of characters in the program code is totalled and compared with the total from the control run If they are different, a change has been made If they are the same, it is likely that no change has occurred

### **OPERATOR PROCEDURES TO CONTROL THE USE OF PROGRAMS AND DATA**

The use of programs and data should be controlled to prevent the processing of wrong versions of programs or data files, to avoid improper actions in handling of files, and to recognize and correct errors The policies and procedures for operating the system and running specific programs should be documented in the operator instructions or the systems and procedures manual The manual should provide the following

- Names of the data files to be run with specific programs (Use of the data files should be restricted to the appropriate programs )
- General control policies, such as the requirements to check external and internal labels of all files used and to restrict computer operation to authorized personnel
- Actions required to respond to system malfunctions, halts, and error messages, including checkpoint and restart procedures
- Distribution procedures for output
- Job control language statements required by the operating system to run each job

The manual is useful both in training new operators and in comparing actual performance with the prescribed procedures. Management should enforce the policies and observe operations on a test basis.

### **LIBRARY SOFTWARE AND OTHER SECURITY SOFTWARE**

The access control features available in many operating systems can greatly enhance control over programs and data. (See chapter 7.) To augment the system software, many data processing installations have written or purchased special library management systems (chapter 5) or special data security software systems. This software keeps track of all the programs and data files that are stored on-line.

The following are major functions of a library maintenance software system:

- To add, delete, or replace programs in the program library
- To facilitate changes to the program code
- To provide reports on program changes
- To restrict the use and changing of programs to authorized individuals
- To provide more efficient storage of files

Although these features are available on some systems, the auditor should recognize that some library management systems may have different features available, because of the type of computer and the particular vendor. Also, the library software may have control facilities that the data processing installation chooses not to implement. The following features of some library maintenance software are particularly useful for control:

- The ability to restrict programmer access to reading and copying those programs or portions of programs necessary to accomplish the assigned tasks
- The production of reports, such as (a) a summary of all changes made to programs in the last month for comparison with a list of authorized changes, (b) a history of all changes made to a program since it was given production status, (c) a detailed report itemizing the source statements that were changed, showing the actual statements that were added, substituted, or deleted, and (d) a list of all programs assigned to a programmer, with current status and activity dates
- Password features to grant access more selectively than under many operating systems
- The ability to encrypt programs and data in storage, thus preventing individuals from using utility programs to bypass controls
- The automatic production of backup copies on a specified schedule or as needed (for example, after every modification of a program)
- Maintenance of a program version number that is automatically incremented whenever a change is made
- Placement and updating of a date of original entry or change on every line of source code to facilitate review of changes

Security software systems protect a variety of computer resources by maintaining a directory of users for the system and allowing them to use only those resources listed in their user profile. For example, only programmers would be allowed to use the interactive programming system, and only operators would be allowed to run production batch programs and access production batch data files. Only authorized users would be allowed to use specific on-line application functions and files.

## 126 Safeguarding Availability, Access, and Use of Computer Facilities

Since data processing management may choose not to implement certain of these features, the auditor should consider the control features that are both present and working in evaluating whether control objectives are met. The presence of library or security software in an installation will not automatically improve control. Library software can facilitate changes; however, if they are not controlled, some of these change features may facilitate unauthorized changes. For example, some library software systems provide a facility to perform temporary one-time changes. The change is made for one run of the program, and the program is returned to its original form. This can be a useful debugging aid for changes, but it can also facilitate commission of fraud (for instance, by running the payroll program once with an unauthorized change and removing evidence of the tampering).

### **Backup, Recovery, and Retention Procedures**

No matter how carefully the computer facilities, programs, and data are protected, physical damage, errors, or irregularities will eventually occur. A basic element of safeguarding these assets is a well planned and well tested backup and recovery procedure. The plan must include all of the elements of the system:

Physical facilities

Personnel (and operating instructions)

Supplies and forms

Application programs, including documentation, and system software

Data, including machine-readable files or source documents needed to recreate the files

The plan should be documented and tested, and copies should be stored at appropriate locations, including an off-site location. (In this way, if a disaster strikes the data processing facility, a copy of the recovery plan will be available.)

The recovery plan should include personnel considerations. Who should be contacted? What are their home phone numbers? Which people are qualified for which duties? How will they be transported to the backup facility? All the individuals assigned responsibility on the recovery plan should be fully aware of their duties.

The plan should also include the required supplies. The programs and data need input and output forms to be effective. The billing program may be useless without forms for the bills. In addition, control over access to the backup supplies, such as check stock, is as important as control over the regular supplies.

### **PHYSICAL BACKUP FACILITIES**

The installation should have arrangements for the use of backup facilities at a different location. They should be located far enough away from the main facilities to not be subject to the same disasters as the main facilities, but they should be close enough to provide quick and reasonably convenient service if needed. The backup facilities may involve the use of the following.

- Manufacturer's installation
- Data processing service center equipment
- Another organization that has the same equipment (perhaps a reciprocal arrangement or cooperative of users of that equipment)



- Replacement equipment from the manufacturer, with prompt delivery guaranteed<sup>6</sup>
- Duplicate facilities at the organization's own backup facility
- Commercial backup facilities

All agreements with outside parties should be documented in a written contract

Seemingly small differences in facilities can preclude processing, so the backup facilities should be tested periodically with actual programs and data.

## **BACKUP OF PROGRAMS**

All software, including systems software, application programs, and documentation, should have backup copies stored off premises. This can be accomplished by renting space in a secure, fireproof, remote location. Some organizations use bank vaults or special data storage service facilities. The special data storage facilities guarantee security precautions, a temperature and humidity controlled environment, and such services as microfilming and twenty-four-hour emergency shipment of stored data.

A less expensive method is to use a different storage location within the same company. One organization that follows this policy delivers copies of its files and changes in documentation each day to another company location, where they are stored in a fire resistant room. Files no longer needed for backup are returned to the computer center for reuse. This method of providing files for remote storage depends in part on the type and portability of the data media used.

## **RETENTION AND RECOVERY PLANS FOR DATA**

The recovery plan should include all machine-readable master files, transaction files, and whatever other data, including source documents, are needed to reconstruct the files.

The data retention plan is necessary for legal and tax reasons, as well as for recovery and audit trail tracing. The Internal Revenue Service's Revenue Procedure 64-12 requires that an EDP system have an audit trail designed so that the details underlying the summary accounting data, such as invoices and vouchers, may be identified and made available to the IRS upon request. The records must provide the capability to trace any transaction back to the original source or forward to a final total. If printouts of transactions are not made when they are processed, the system must have the ability to reconstruct these transactions. In Revenue Ruling 71-20, the IRS states that cards, tapes, disks, and other machine-readable media are the equivalent of manual ledgers and journals and, therefore, should be retained for the same length of time ledgers and journals are retained.

Because the retention plan is affected by the characteristics of the media involved, this discussion considers source documents, punched cards, magnetic tape files, disk files, and dumps (copying of contents) to other media.

### **Source Documents**

The source documents on which an input file is based must be retained intact until the file is proved and balanced with its controls. At this point, the documents may be filed or otherwise disposed of, unless there are legal or other reasons for their retention.

---

<sup>6</sup> In this case the plan should consider a backup location for the replacement equipment when it arrives

## 128 Safeguarding Availability, Access, and Use of Computer Facilities

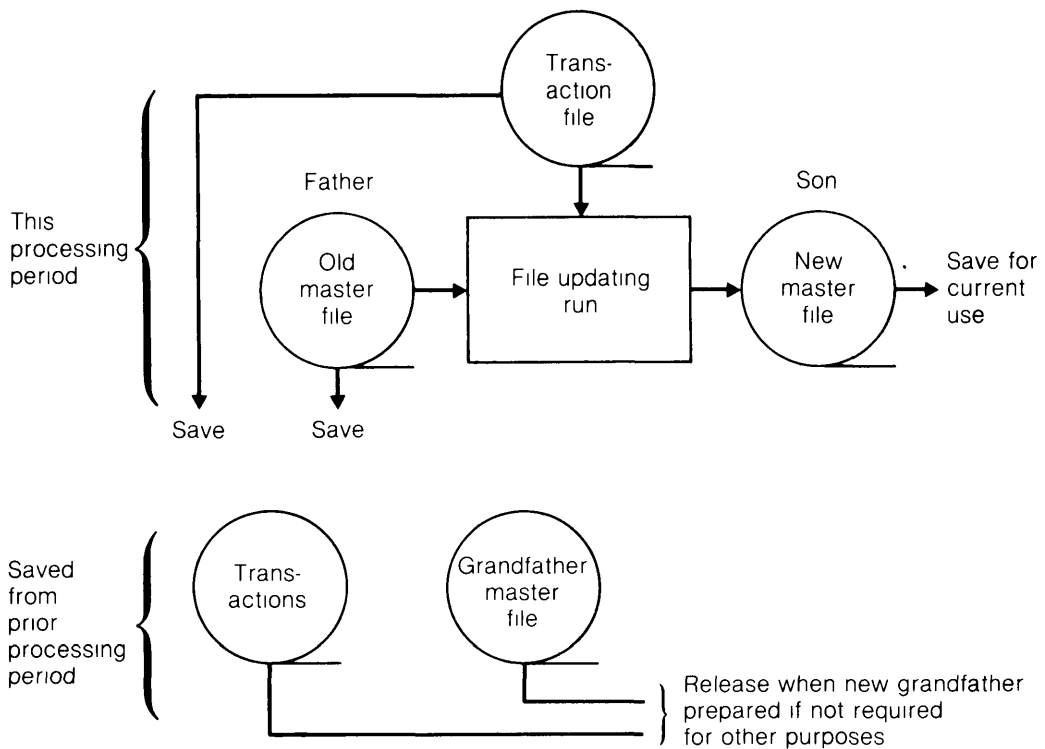
A company that intends to produce records on microfilm directly from the computer, off-line from magnetic tape, or by photographing original documents or printouts should review IRS Revenue Procedure 75-33 and Revenue Ruling 75-265. They state that the micrographic system must meet certain specific criteria and that a description of the system must be submitted to the IRS for approval.

### Magnetic Tape Files

Backup support for tape files is usually provided by the son-father-grandfather concept (figure 8-6). An organization produces an updated master file at each processing by reading the master file from the previous period and incorporating any changes before writing the new file. The following is an example of the normal backup available for daily updating of the file after processing the transactions of any given day (Wednesday in this case). Keep in mind that the processing creates a new tape file but does not destroy the old one.

- Wednesday's file (son)
- Tuesday's file (father)
- Monday's file (grandfather)

If Wednesday's tape is destroyed during Thursday's processing, Tuesday's tape is processed again with Wednesday's transactions in order to recreate Wednesday's tape. This type of backup is basic to all magnetic tape processing.



**Figure 8-6** Grandfather, father, and son backup for magnetic tape files

If files are to be re-created under the son-father-grandfather concept, the transaction records used to update the original files must also have been retained. In the example, Tuesday's and Wednesday's transaction records must be available, since file re-creation is accomplished as follows:

Monday's master + Tuesday's transactions = Tuesday's master

Tuesday's master + Wednesday's transactions = Wednesday's master

In the son-father-grandfather concept, the grandfather tape may be released when a new grandfather tape is produced. Thus, two backup tapes plus transaction tapes are held at any particular time. Sometimes, a great-grandfather may also be saved, but this is not usually necessary.

There are other considerations in the establishment of a retention schedule for magnetic tapes. End-of-the-month files and other critical records may be saved to preserve data records, which are likely to be needed for the preparation of special analyses. Once an old backup master file is released, its associated transactions can also be released.

### Disk Files

In disk file processing (whether on traditional or floppy disks) the old record is altered by updating. When a file is updated, a record is read into storage, altered, and copied back onto the disk in the same location, erasing the old record. Unlike magnetic tape processing, disk file processing does not automatically produce a backup copy. To obtain a backup copy, a special copying procedure must be performed. Backup for disk files can be created by copying the data to another disk or by copying the data to another file medium (figure 8-7).

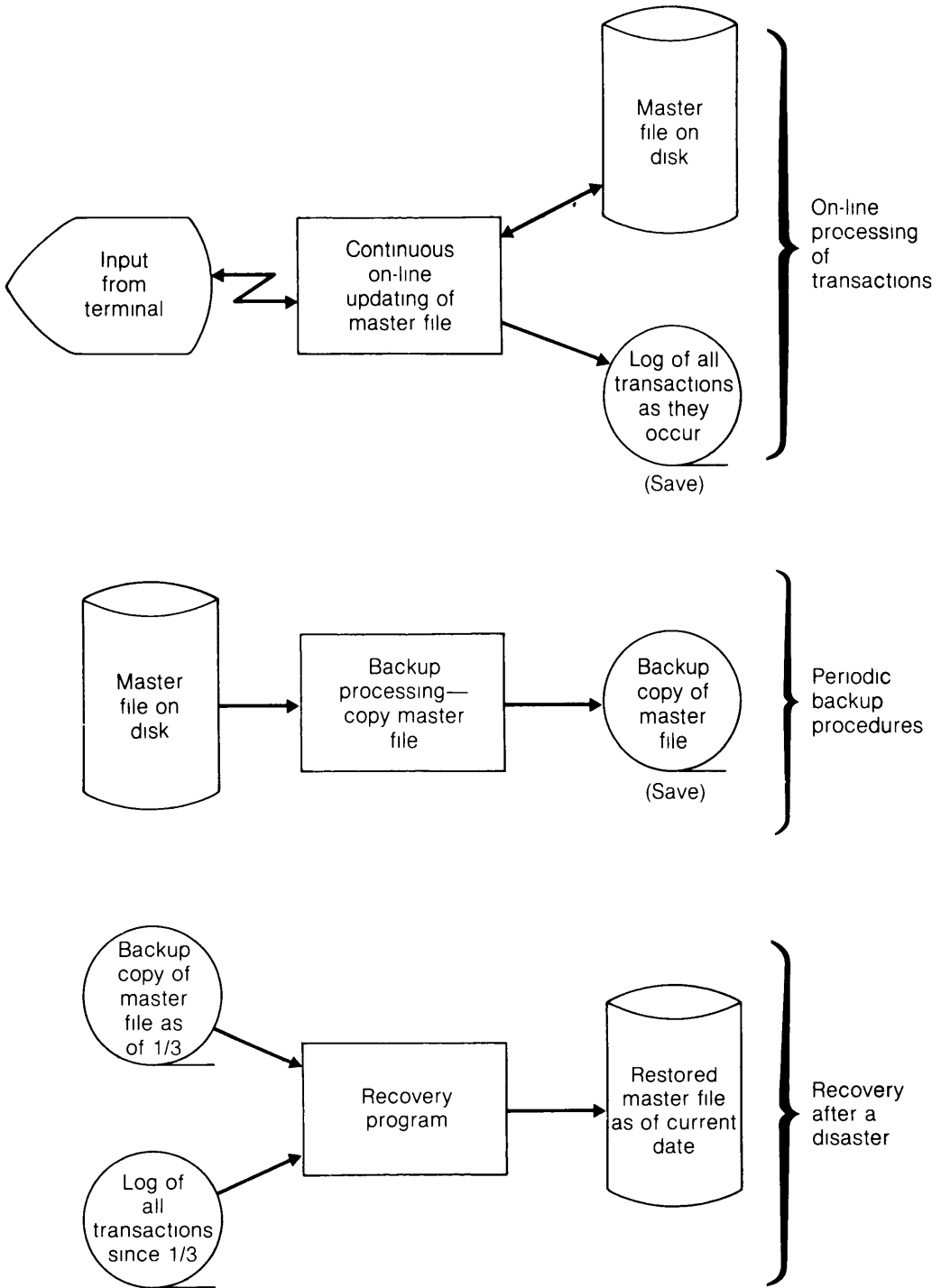
In most cases, daily duplication of the disk, although requiring time and expense, is justified by file retention requirements. As an alternative, a weekly or biweekly disk duplication schedule, coupled with a plan for retaining intervening transactions, may provide adequate support. Since this alternative makes considerable reprocessing necessary before a duplicate file is available, it may not be satisfactory if rapid file replacement is essential.

Requirements for transaction retention in disk systems are similar to the requirements in tape systems. Basically, transactions must be retained until the related files have been proved and until they are no longer required for backup or other processing needs.

### Dumps

One way to retain a disk file is to reproduce (dump) it on another file medium. A magnetic tape drive attached to the computer system provides an effective form of disk file backup. The disk file can be dumped onto magnetic tape in a matter of minutes, and the tape file can then be given fireproof, off-premises storage. This combination of disk file with tape support makes a powerful system of file control, and it is often used when there is only a single disk pack drive or fixed disk units. Magnetic tape files are less expensive than disk files, however, addition of the tape unit and tape controller may increase the system cost. Streaming tape units and floppy disks are often used for low-cost backup for rigid disk files.

To serve as backup for a disk file, the magnetic tape dump or other media dump of the file should receive protected or off-premises storage until they are no longer required for backup or other needs.



**Figure 8-7** Backup and recovery for on-line disk files

## Summary

To ensure the continued, authorized processing of data, the computer facilities, programs, and data must be protected from physical damage and unauthorized access and use.

Safeguarding of the computer facilities includes the location and design of the computer room, fire detection and fire-fighting techniques, adequate insurance, good housekeeping and preventive maintenance, physical safeguards for files, and controlled access of personnel to the computer room

Computer programs and data should be protected from unauthorized use or changes. This can be accomplished by assigning librarian personnel responsibility for the safeguarding of files, using techniques such as passwords and encryption, providing controls over using or altering of programs and data, and using library software to keep track of file access and usage.

Backup, recovery, and retention procedures are vital to ensure the continued functioning of the system if problems arise. The procedures should encompass all the elements of the system, including hardware, personnel, supplies, programs, and data.

The controls and safeguards described in this chapter include both accounting controls, affecting the reliability of financial records, and administrative controls, affecting the continuation of data processing capabilities.

Accounting controls include the following:

- Librarian function
- Passwords and other identification techniques
- Encryption
- Program change process
- Procedures to control the use of programs and data (manual and automated)
- Control of access to the computer room
- Physical safeguards for files

These procedures and control techniques are part of accounting control because they control access to the computer facilities, the programs, and the data and because they control changes made to programs and data. The programs are processing procedures and should be used or changed only as authorized. The data records must be protected from unauthorized additions, changes, or deletions, since any of these actions, if unauthorized, can affect the financial statements and the accountability for assets.

Administrative controls (continuation provisions) include the following:

- Location and construction of the computer room
- Fire detection and firefighting
- Insurance
- Housekeeping and preventive maintenance
- Physical backup facilities
- Backup programs and procedures
- Retention and recovery plans for data

The continuation provisions are directed at ensuring the continued availability of computer data processing capabilities (facilities, hardware, software, and data). They can have a significant impact on future preparation of financial statements, and many auditors include them in the agreed-upon scope of the financial statement audit.



---

## SECTION III

# Processing Methods and Controls for Applications

Applications are uses to which computer data processing activities are applied. The uses are associated with organizational functions and objectives. In the following examples, the first two applications are in support of major organizational functions—obtaining payment for goods and services and paying for employee services—and the third is in support of decision-making to select among alternative projects or investments.

<u>Application</u>	<u>Description</u>
Accounts receivable	Process all transactions affecting the accounts receivable balances and prepare documents and reports related to accounts receivable.
Payroll	Process all payroll transactions and prepare documents and reports for payment and reporting purposes.
Investments analysis	Process data on alternative projects or investments and rank them according to management criteria.

The term “application” can apply to major processing applications or to component activities. For example, the accounts receivable application may be divided into several component applications:

- Process accounts receivable transactions
- Customer billing
- Analysis and aging of accounts receivable

The term “system” is often applied to applications. These are application systems because the processing activities take inputs, process them, and produce outputs.

A useful classification of application controls for audit purposes is—

1. Input controls, including data preparation and terminal entry
2. Processing controls
3. Output controls

## 134 Processing Methods and Controls for Applications

Four areas of concern that may affect more than one of the categories of input, processing, and output controls are these

1. Authorization
2. Data validation
3. Error handling
4. File quality maintenance

Another area with audit significance is the audit trail, which is essential for management purposes and important to the performance of audit procedures

Section III describes the processing methods and controls for applications.

Chapter 9—Control Implications of Data Processing Methods

Describes different application processing methods and notes control implications

Chapter 10—Application Controls Authorization, Data Validation, Error Handling, and File Quality Maintenance

Describes application-specific methods and generalized approaches to authorization, data validation, error handling, and file quality maintenance

Chapter 11—Application Controls Data Preparation, Input, Processing, and Output

Describes the application controls to prevent, detect, and correct errors as the application transactions flow through the procedures for data preparation, input or direct entry, processing, and output

Chapter 12—The Audit Trail in Computer Data Processing

Describes the purpose of an audit trail or processing trail, the requirements that need to be met, and alternative procedures for meeting the requirements



---

# Control Implications of Data Processing Methods

# 9

Computer-based data processing may be performed in a variety of ways. The processing method or approach affects the audit because it affects such factors as the complexity of the system, the internal accounting controls, the visibility of the audit trail, and the ease of record retrieval. These in turn affect the application of audit procedures and the availability of audit evidence. There is no generally agreed-upon classification of computer data processing methods. The classification used in this chapter is oriented toward auditor use.

## The Basic Computer Data Processing Cycle

There are two fundamental variations for the computer data processing cycle. Alternative 1 is the traditional cycle in which documents are used to capture source data that are then prepared for input to the computer, alternative 2 is a cycle in which source data are entered directly by a terminal to the computer or to a magnetic storage medium (figure 9-1).

### Alternative 1—Traditional Cycle

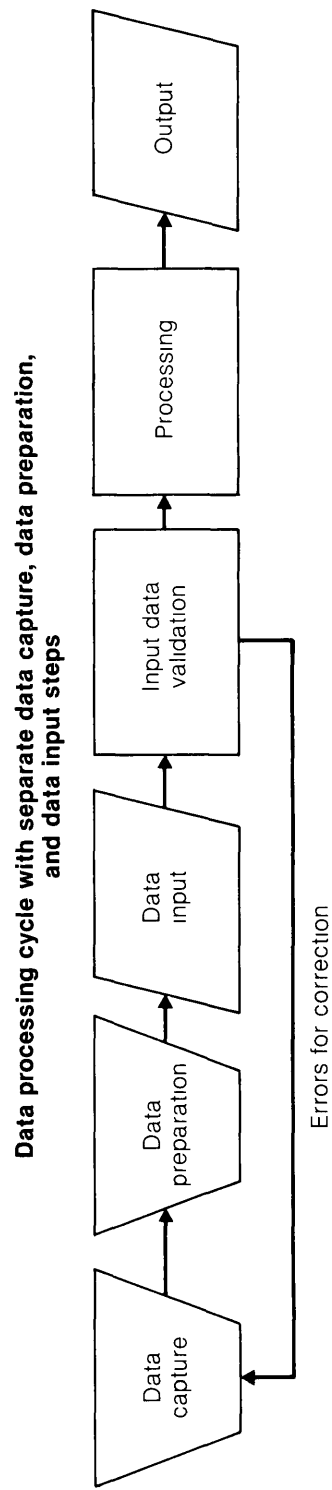
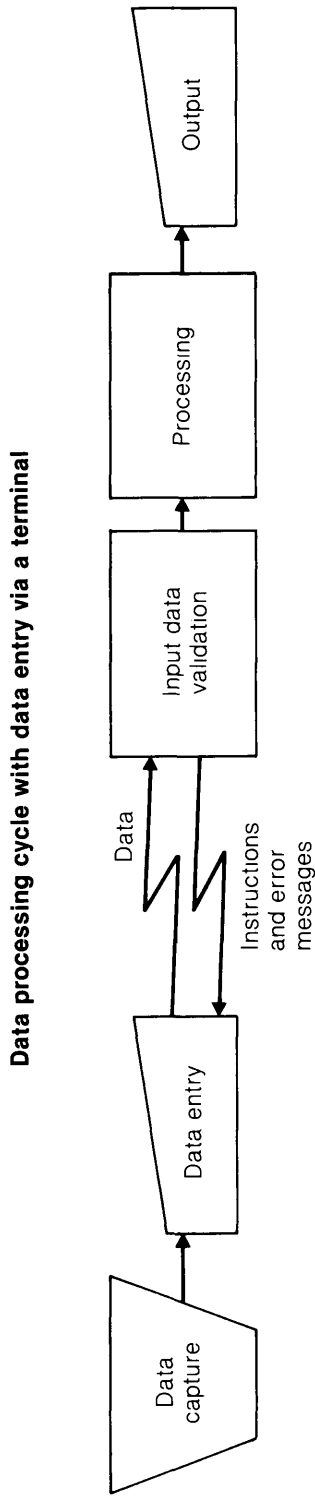
- 1 Data capture
- 2 Data preparation
- 3 Input
- 4 Input validation
- 5 Processing
- 6 Output

### Alternative 2—Terminal Entry

- 1 Data entry
- 2 Input validation
- 3 Processing
- 4 Output

## DATA CAPTURE AND DATA ENTRY

Data capture consists of activities to obtain data items and record them for subsequent processing steps or other use. Data entry is the direct input of data to the computer or to a magnetic storage medium. Data entry may be performed as part of data capture, or it may be performed as a separate step, using documents prepared by data capture activities.



**Figure 9-1** Two basic data processing cycles

There are three different approaches to data capture. Data items may be recorded on documents in a form that is not machine-readable. If data items are recorded in this way, they must then be prepared as machine-readable records before they can be entered for computer processing. In the second approach, data items can be recorded on documents in a machine-readable form, bypassing part of the data preparation phase. In the third approach, data can be entered directly to the computer or to a magnetic storage medium, eliminating all data preparation and input activities. This method can be called "terminal data entry."

Data capture via written notes and filled-in documents requires subsequent data preparation to transcribe the data into machine-readable records. Examples of this form of data capture are manually prepared employee records, manually recorded time sheets, and handwritten sales orders.

Data capture in machine-readable form requires the initial recording of data in a form that can be read by an input device. This can be performed at data capture or by activities in a prior application. An example of the latter method is the use of machine-readable turnaround documents prepared by a billing application that, when returned by a customer along with payment, provides data capture in machine-readable form. The following are some machine-readable recording methods:

- Handwritten numbers to be read by an optical character reader, such as the handwritten recording of meter readings for an electric utility
- Handwritten marks to be read by an optical reader, such as a machine-scored examination
- Documents typed in a machine-readable font, such as a credit document typed by a credit interviewer
- Punched card as turnaround document

Documents for capture in machine-readable form have the characteristics of ordinary data capture documents, but they must be prepared with greater care and precision. The use of documents prepared in this way eliminates the need for a separate step to convert data to machine-readable form.

Terminal data entry at the transaction point uses a typewriter or visual display unit (a CRT or cathode ray tube device). The terminal may be on-line, connected to the computer either directly or via data communications, or it may be off-line, with logic capabilities that can be used to receive, check, and record the transactions. An example of an application that enters data through a terminal for subsequent processing is a retail store data entry and checkout system. Terminal data entry may be used to capture and enter individual transactions as they occur, or it may be used to enter data from documents. Data from documents can be entered immediately as they are received by the terminal operator, or documents may be batched and control totals prepared prior to entry.

The hardware and software and data communication (if it is used) that support terminal data entry are expensive. However, terminal data entry provides the following advantages:

- Data items are in machine-readable form after entry
- Data can be entered while a transaction is being performed. Computer responses to data entry can prompt entry of complete, correct data. However, if periodic entry of batches of data records is more efficient, this can be done instead.
- There is immediate data validation and error feedback so that the individual entering the data can make immediate corrections.

### DATA PREPARATION

Data preparation (or data conversion) applies to the traditional batch cycle in which four types of activities may be used to convert data recorded on documents into a machine-readable form for computer input. The extent to which these activities are required depends on the technology used in data capture. For instance, did the data capture produce documents that are in machine-readable form? Are the documents prepared in a form that can be read into the computer quickly and efficiently? The data preparation activities are as follows:

- Manual review of source documents and, if necessary, correction, addition, or deletion of data
- Preparation of processing controls, such as assembling data into batches and establishing batch control totals
- Transcription to machine-readable form, verification of the correctness of the transcription, and validation of some data items
- Conversion from one machine-readable form to another

The manual review of data prior to processing may be used to add codes or other data as well as to make sure that the data are complete, legible, and reasonable. Documents are assembled in batches, and control totals are prepared. Examples of control totals used at this point are transaction record counts, totals of financial data fields, and hash totals of fields. The control total is written on a control document or ticket, recorded in a log, and attached to the batch of documents. A control advantage of this procedure is the segregation of the function of preparing control totals from processing functions.

The batches of transaction documents and their accompanying control documents are transcribed into machine-readable media. The transcription operation may involve the following steps:

- Keying onto disk or tape storage
- Keying at a terminal to enter data onto remote disk or tape storage
- Typing documents for optical scanning
- Handwriting data onto forms for optical scanning
- Punching cards

Some of the methods of transcription to machine-readable media use a verification operation that is based on dual entry of the data, other methods require visual proof-reading. In the case of keypunches or keydisk units, the equipment can do a validity check on a coded data item containing a check digit. Keydisk units and some keypunches can also make additional checks on the data fields. The extent of checking of data transcribed by entry at a terminal depends on the logic capabilities of the terminal. Many of these keying devices contain features that assist in the verification of control totals, for example, they accumulate control totals for different fields in the keyed-in records, such as account numbers, employee numbers, and hours.

In those cases in which data capture or data preparation results in machine-readable data on a medium that has a slow speed or is difficult to read, data preparation may involve a conversion to a more accessible or faster storage medium. Examples of such conversion are the transfer from keydisk storage to magnetic tape for use

in input, the reading of documents by an optical character reader and their transfer to magnetic tape for ease of operation and input speed, and the transfer of cards to magnetic tape for greater input speed and ease of operation

## **INPUT**

In the input phase, the data from the preparation phase is read and stored for processing. Input validation is generally an integral part of terminal data entry, but it will be considered separately because it can be performed independently. If data validation tests are performed prior to input, input procedures should include a check of the control totals developed during the data preparation phase.

## **INPUT VALIDATION**

Input validation, sometimes called editing, examines data items at time of entry or during input and performs various tests to detect errors. Control totals may be generated during data validation or checked if they were generated in a prior stage. Validation is part of both the traditional cycle and terminal entry data processing. Partial validation may be performed by a keydisk unit or an intelligent terminal used for data capture or data preparation, with a complete validation performed at input. The following are some major validation tests:

- Use of check digits to validate account numbers
- Tests of codes using a table of acceptable values
- Tests of data for reasonableness (range tests)
- Tests for missing fields
- Tests for alphabetic or special characters in numeric field
- Comparison with file data

The last of these tests, comparison of data items with file data, is impractical in certain systems, it can be performed only if the file records can be accessed during the validation process. In general, the scope and number of validation tests depend not only on the technology of the controls but also on their cost.

Data items tested and found to be erroneous are rejected. If batches of data are being validated, it is common to write any errors on an error file and send the error records back to the data originators for correction and re-entry. Batch control totals are adjusted for the rejected items.

Performing validation at the time the operator enters the data to the computer or at an intelligent terminal allows erroneous data to be identified immediately at entry. An error message is sent to the entry terminal so that the operator can correct or re-enter the item. The corrected data is then revalidated. The data that passes the entry validation test continues into processing.

## **PROCESSING**

Three major processing operations are performed with input data: sequencing, transaction processing, and file updating.

## 140 Control Implications of Data Processing Methods

Sequencing, through sorting and merging, may be performed prior to transaction processing or file updating, it may also be performed after processing but prior to output in order to have the output in an appropriate sequence

Transaction processing operates with input data to produce a transaction document, a response at a terminal, or some other transaction record. For example, information about hours worked may be used to produce a paycheck.

File updating may involve additions, corrections, and deletions to the file, as well as updating of file data using values from transaction data. A transaction may, therefore, be used both to prepare a transaction output and to update a master file record. Thus, a payroll transaction may result both in a paycheck and in the updating of the master file record for the employee, for example, updating of wages to date, FICA to date, and so on

### **OUTPUT**

Processing output can be an updated master file, a temporary file to be held for future processing, output files, or direct output to a printer or visual display terminal. The output files are generally printed as transaction documents or reports, but other media may be used, such as a punched card, a turnaround document, or microfilm. In large computer installations, output is not printed directly but is written, or spooled, onto a special output file to be printed when the line printer is available.

## **Classification of Computer Files**

The organization of computer files is an important factor in computer processing methods. File organization is based on the needs of each application and the physical characteristics of the file storage device.

### **FILE STORAGE DEVICES**

Storage is frequently classified as primary (internal) storage or secondary (auxiliary) storage. The active portion of the operating system and programs being executed are in internal storage, files in use and programs not currently in use are in auxiliary storage. The most common internal storage uses semiconductors or magnetic cores, the most common secondary storage media are magnetic disks and tapes. Data are stored on a tape or disk by recording patterns of magnetic polarization on small sections of a ferrite-coated surface. Data are stored at a specific location on the medium, no other data can be stored at that location without destroying the data already recorded there. However, data can be read without altering the magnetic recording.

Locating stored data can be performed serially or directly. Serial, or sequential, access is commonly associated with magnetic tape because the tape is read serially from beginning to end. The first record must pass under the read/write head before the second record can be accessed, the second record must pass before the third, and so on. In order to access the last record on the file, all previous records must have been read.

In direct access, or random, storage, such as disk storage, the storage medium revolves, and the read/write arms move in and out so that any location is accessed directly by the combination of their movements. In an alternative disk storage design, there is a read/write head over every recording track, making the movement of read/write arms unnecessary. If desired, locations on a magnetic disk can be read serially as well as randomly.

## FILE ORGANIZATION

The major methods of file organization are (1) sequential and direct access, used in data processing, and (2) lists, used primarily in retrieval applications. In sequential organization, the records in the file are sequenced in order by a record key, such as employee number, customer identification number, or a social security number. The record with the lowest number key appears in the first location on the file, the record with the next-lowest number key appears next, and so on. This organization is easy to use for magnetic tape processing, but it is also used with magnetic disk storage. In processing using sequential file organization, it is not necessary to specify a storage location address to obtain a record. Since the records are stored in order, a record is located when its place in the sequence is found.

In direct organization, a record can be located without considering its sequence. Direct access organizations can be implemented in a variety of ways, the two most common being random procedure and index methods. In the random procedure, the record key is manipulated arithmetically to produce a random number that identifies a storage location. If a second record is assigned to the same location using this procedure, one of them is stored in an overflow location, and a procedure is established to find the overflow record during processing. In the index method of organization, an index is established that indicates the storage location for each record key. When information is required, the index is searched for the record key to obtain the record storage location.

Another type of organization is indexed sequential, which stores records sequentially but allows them to be located directly through indexes. This makes possible the use of either file order processing or transaction order processing.

List organizations are fairly complex and are used primarily in retrieval applications when it is helpful to know the relationship among various records. List organizations may use pointers (data fields containing addresses of other records) to link records together. Another list approach employs indexes that identify all records having given attributes.

## DATABASE AND DATABASE MANAGEMENT SYSTEMS

As explained in chapter 7, separate applications may use separate files that contain duplicate data, but processing efficiency and improved data control are achieved by creating a common file for all applications using common data. In addition, lookup and retrieval operations are simplified and improved. However, the common files, called a database, require strict control over access and changes to the data.

The database management system provides data requested by an application program and updates data items by using current information provided by an authorized application program. It removes most of the file access and file update logic from the application program.

## MULTIPLE-COMPUTER ACCESS TO FILES

An organization that has computers and files at several locations can interconnect the computers via data communications to form a distributed data network. In another kind of distributed data network, the computers of one company can access the computer and files of another company. For example, airline reservation computers make reservations via data communications directly to connecting airline computers. Such networks are complex communication systems, however, the basic entry, data validation, and processing steps must still be followed.

## Processing Methods

Computer data processing literature has traditionally identified two major methods of processing, on-line and batch. In on-line processing, individual transactions are processed immediately; in batch processing, batches of transactions are processed periodically. Computer technology has developed alternative processing approaches that do not fit these traditional classifications. An alternative classification is based on the separation of the transaction entry processes from the processing of the data.

Transaction entry processes can involve either single transactions entered immediately at a terminal or batches of transactions periodically prepared for processing by one of several methods of data preparation. Transactions that are entered immediately via a terminal need not be processed immediately. A general advantage for immediate transaction entry procedures is the validation of entries by the computer logic. If an error is discovered, an immediate response to the entry terminal allows the operator to correct the transaction. On the other hand, data preparation can, in many cases, be performed very efficiently on batches of transactions.

Processing, also, can involve either single transactions processed immediately or batches of transactions processed periodically. Periodic processing is efficient and is often very consistent with the nature of the activity for which processing is performed, for example, weekly payrolls, monthly billing, or the daily production schedule. However, there are also many activities that can benefit from on-line, immediate processing of transactions, for example, reservations, commitment of inventory to a purchaser, or inquiry about the status of a production order.

Altogether, there are three major processing methods for computer data processing applications.

1. Terminal Entry/On-line Processing. Immediate terminal entry of single transactions and immediate processing of single transactions. This is commonly referred to as on-line, real-time processing.
2. Terminal Entry/Batch Processing. Immediate terminal entry of single transactions, which are stored for periodic processing of batched transactions.
3. Batch Entry/Batch Processing. Periodic preparation of batched transaction documents and periodic processing of the batched transactions.

There are many variations within these groups. Common variations are listed in table 9-1, in descending order of complexity.

### TERMINAL ENTRY/ON-LINE PROCESSING

The terminal on-line method is the most complex of the three methods because both entry and processing are immediate. Terminals are used for input and output. Transactions are validated immediately, and a message is sent to the terminal if the transaction is not valid. Necessary reference data items (in files or database) are available to the transaction validation process and the transaction processing.

Transaction preparation and control of entry errors can be improved by the immediate validation process, however, the traditional batch control totals prepared prior to data preparation are not available when individual transactions are entered immediately. Control procedures for on-line processing generally require that each transaction be assigned a unique identifier (usually by the program) when it is entered. This provides an audit trail based on individual transactions.

A major consideration in on-line processing is the handling of the reference data (in a master file or database). The different methods of handling reference data lead to the



- |   |
|---|
| <ul style="list-style-type: none"> <li>I. On-line terminal entry of single transactions and immediate processing of single transactions (terminal entry/on-line processing) <ul style="list-style-type: none"> <li>A. Immediate, complete processing of all transactions, including the updating of data in master files or database</li> <li>B. Immediate processing of all transactions with concurrent updating of data in a memo file or changes file and periodic batch updating of the data in the permanent file</li> <li>C. Immediate processing of transactions with only periodic updating of reference data</li> <li>D. Immediate processing of transactions not requiring reference data in file or database</li> </ul> </li> <li>II. Terminal entry of single transactions and periodic processing of batched transactions (terminal entry/batch processing) <ul style="list-style-type: none"> <li>A. Immediate validation of transactions by computer or intelligent terminal with entry and inquiry program having access to reference data in master file or database</li> <li>B. Immediate validation of transactions by terminal program or computer program without use of reference data</li> </ul> </li> <li>III. Periodic preparation of input transactions and periodic processing (batch entry/batch processing) <ul style="list-style-type: none"> <li>A. Transaction order batch processing</li> <li>B. File order batch processing</li> </ul> </li> </ul> |
|---|

**Table 9-1** Processing methods in descending order of complexity

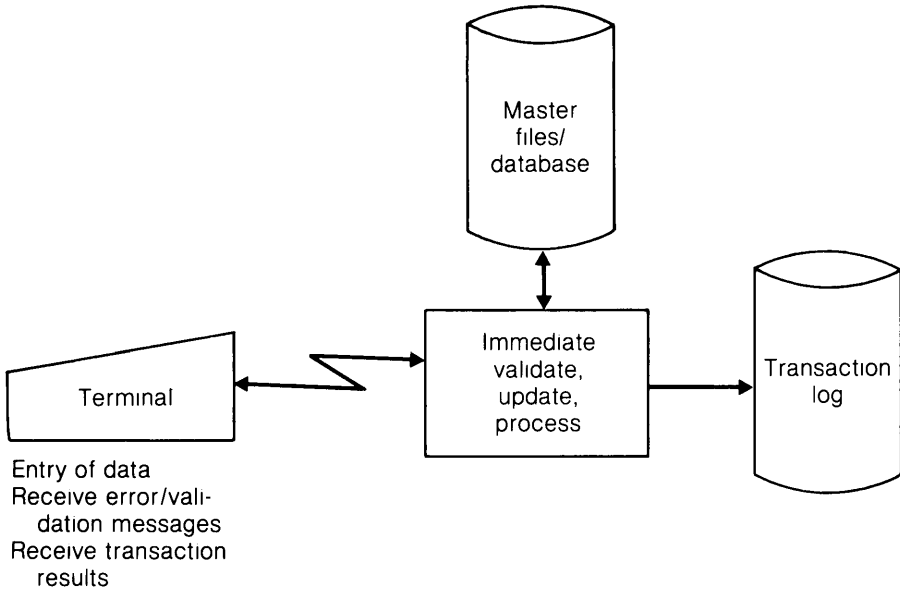
design variations for on-line systems listed in table 9-1. The reasons for use, examples, and advantages and disadvantages of each are summarized in table 9-2

The most complete and most complex method of on-line processing is immediate processing of all transactions, including transactions that update the files associated with the application. An example is an on-line inventory system that updates the inventory file with receipts, shipments, new items, deleted items, and so on. The advantage of the method is that the file is kept up to date, the disadvantage is the complexity of handling all transactions on-line. For example, a disbursement of a new inventory item may be entered prior to the transaction to establish the item as a stockkeeping unit. Errors may enter the file and be difficult to back out since they may cause other transactions to be initiated, as, for example, an inventory reorder caused by an issue from stock. This method of processing is diagrammed in figure 9-2. Because of the possibility of either major processing errors or hardware failures with uncertain results on files, a computerized log should be made of all transactions, and a backup copy of the file should be made as well. Using the beginning-of-period file plus the logged transactions, the file may be reprocessed to recover from the errors.

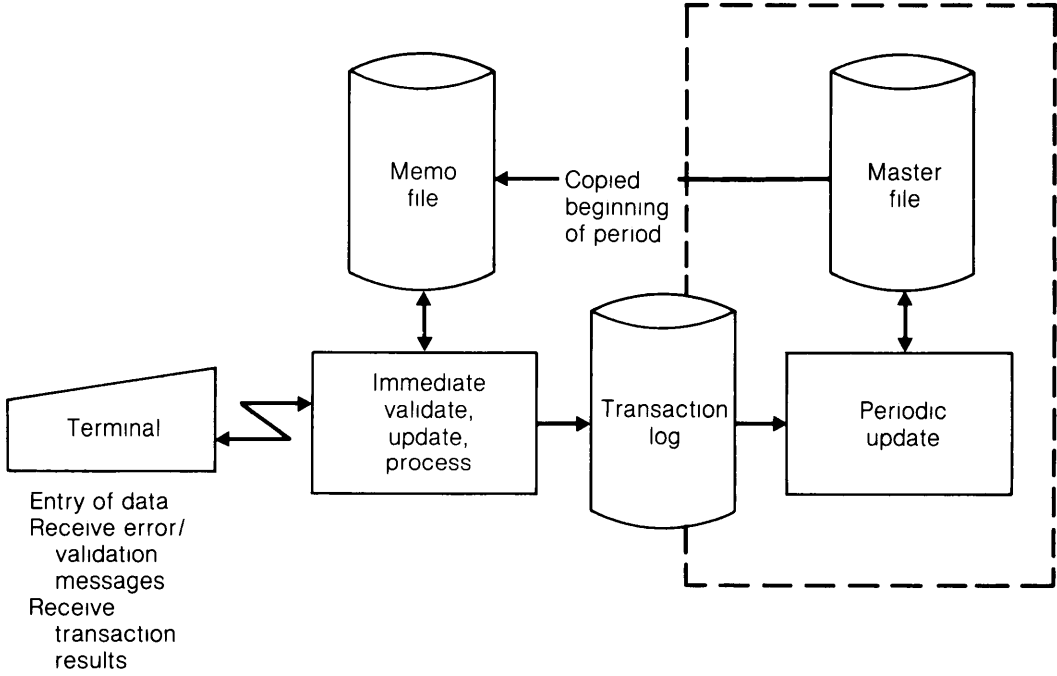
The benefits of completely on-line processing are significant, but the risks from on-line updating of errors or irregularities may be too great. Alternatives are to process on-line but update only a copy of the master file or do pseudo-on-line updating. In memo updating, the updated memo file is used only during the period in question (for example, a single day). A copy of the transactions is kept, and the permanent file is updated in batch processing (for instance, at night). The updated file is then copied as a memo file to start the on-line processing for the next period. This variation of on-line processing is diagrammed in figure 9-3. A similar alternative is to perform pseudo-updating of the master file. In this method, all changes to the master file are kept in a temporary changes file. Prior to any transaction requiring up-to-date data, the master file changes file is searched for changes. If there are changes, they are reflected in the data provided for transaction processing. The permanent updating of the file is done periodically.

Immediate Processing With Alternative Use/Updating of Data in Master Files or Database	For Each Alternative			
	Reason for Use	Example	Advantages	Disadvantages
Immediate updating	Need for up-to-date file	Airline reservation system	Permanent file always up to date, no extra processing	Error in updating difficult to detect and correct
Immediate memo or pseudo updating with periodic permanent updating	Need for up-to-date file but additional safeguards on updating	Financial transaction with complex relationships	Provides protection against effects of updating	Extra processing required for permanent updating
Periodic updating of reference data	Access needed but immediate updating of reference data not required	On-line warranty claims file	On-line access with simplest updating	Access to a record may not obtain latest information
No use of reference data in file or database	No need for reference data	On-line rate-of-return analysis	—	—

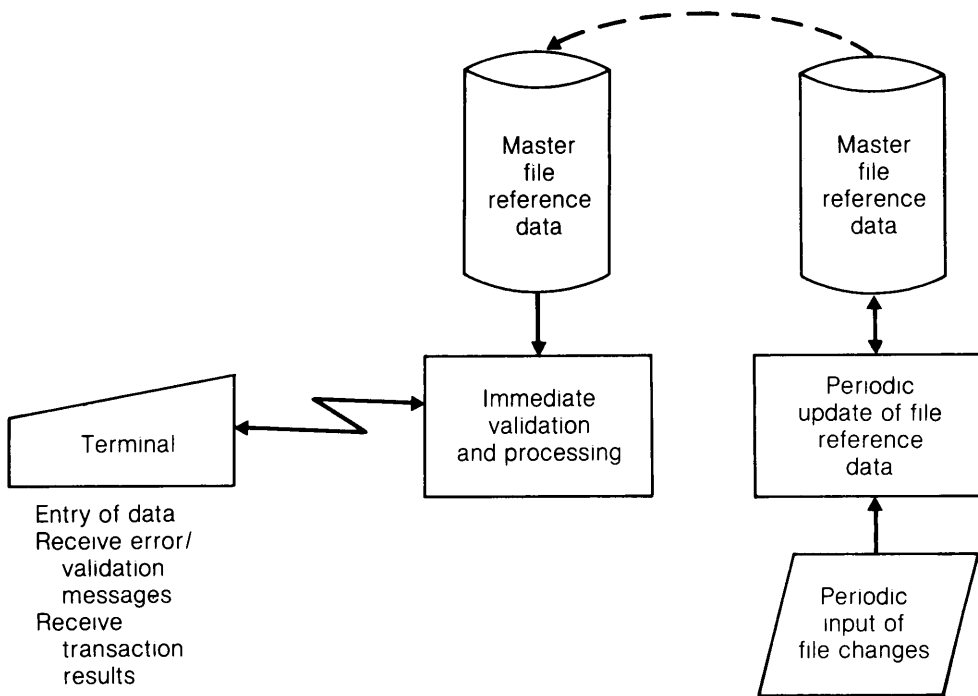
**Table 9-2** Alternative designs in on-line entry/on-line processing



**Figure 9-2** Immediate processing of all transactions



**Figure 9-3** Memo updating



**Figure 9-4** Immediate processing with periodic updating of file

For some applications, updating of file data is not critical during a period. For example, an application to process warranty claims on-line may have a file of past claims for each unit to use in processing a claim request. The file of past claims must be accessed on-line but not necessarily updated immediately. This variation is illustrated in figure 9-4.

Some on-line applications do not require any file for reference or updating. For example, an on-line application to perform rate of return analyses or to prepare depreciation schedules for a new piece of equipment probably does not need any data beyond the values stored in the program or provided at input.

### TERMINAL ENTRY/BATCH PROCESSING

In this method, single transactions are entered immediately via a terminal. Each transaction is validated immediately, with error or reject responses returned to the person entering the transaction. Those transactions that are accepted as valid are written onto a computer file to be processed as a batch in a periodic batch processing run. The method retains the advantages of on-line entry and validation but uses less complicated periodic processing methods.

The on-line entry/batch processing method is most appropriate for applications for which data can be entered and validated on-line but for which the transaction results are not needed immediately. The validation program may be operated by the main computer, by a minicomputer, or by an intelligent terminal. Two variations of this method are (a) immediate validation of transactions by computer logic with entry and inquiry programs that have access to reference data from the master file or database

and (b) immediate validation of transactions by computer logic without use of reference data in a master file or database.

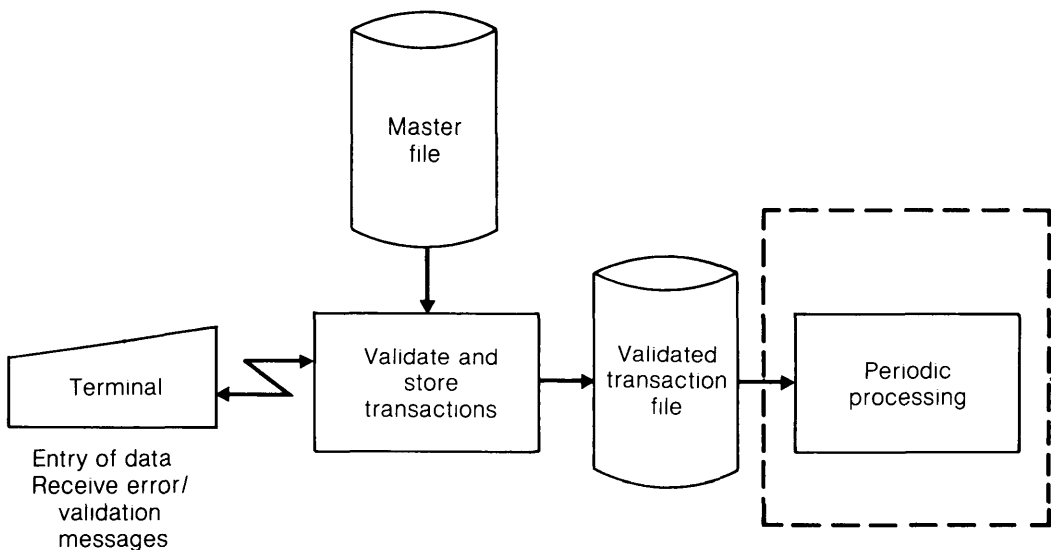
The first alternative provides the most complete validation because it allows reference data to be obtained for use in validating a transaction. The reference data can also be used to assist in formulating the transactions. If the entry program is in the main computer, which also keeps the relevant files, the application merely accesses the files. If the entry logic is contained in a minicomputer or intelligent terminal, the data needed for the entry program may be copied and down-loaded from the main computer. This method is illustrated in figure 9-5.

An example of an application that might be designed in this way is a program to apply payments received to outstanding invoices in an open-item accounts receivable system. In order to apply the payments to a customer account, it is necessary to have a list of outstanding invoices from the customer records in the accounts receivable file. On-line access is therefore important in formulating the transaction to apply the payment, but the updating of the open invoice file need not be done on-line. The input transaction can also be validated using data in the customer record.

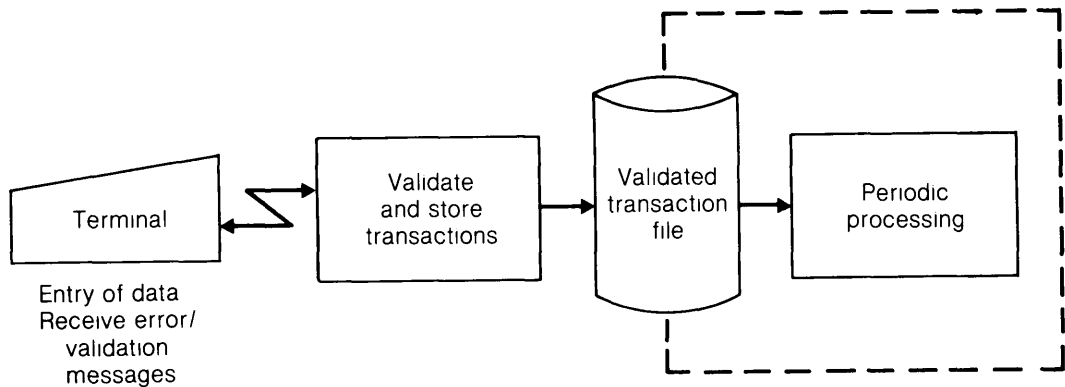
In some cases, the entry logic may not require access to the related file record, or it may not be worth the cost of file accesses. In such cases, a validation program using stored values applicable to all transactions of a given type is used to validate input transactions. The valid transactions are then written on a file for subsequent batch processing (figure 9-6). For example, payroll data may be validated by comparing job numbers and department numbers with a table of valid numbers. Total hours per period are compared with reasonable limits.

It is, of course, possible to store unvalidated data that have been entered via a terminal on a transaction file. Such an option may be useful in a special case, but it generally has very little advantage.

The terminal entry/batch processing method has important control advantages. It provides the control advantages of on-line entry, with immediate validation and imme-



**Figure 9-5** Immediate validation with access to master file



**Figure 9-6** Immediate validation without use of reference data

mediate correction by the person who entered transactions, and it provides the control advantages of batch processing, with pre-established control totals, batch listings, and a controlled sequence in which transactions are processed. Its main disadvantages are not related to control and audit but to the lack of immediate processing and lack of up-to-date files.

### **BATCH ENTRY/BATCH PROCESSING**

The traditional batch processing approach prepares a batch of data for input into processing, using a number of the following methods:

Keying to disk (keydisk)

Keying to diskette or tape cassette

Use of turnaround documents read by a card reader or optical reader

Use of documents that can be read optically

Keying to cards

The batch method uses resources efficiently because similar transactions are processed together. Control totals can be established prior to processing. The batch number is very useful as a processing reference. The major disadvantages of the batch method are the delays in updating records and processing corrections. Errors that are detected are returned to the original source for correction, but this takes some time.

Two major forms of batch processing are transaction order batch processing and file order batch processing. The major difference between the two is the flow of work. In file order processing, transactions are sorted into the same order as the file before they are processed; in transaction order processing, this step is not performed. Transaction order processing may be effectively used for applications having relatively few transactions and a file organization supporting direct access. For example, updating transactions in a purchase order file to post receiving data may be processed most efficiently in the order the reports were received even though the file may be arranged by purchase order number.

In file order processing, also called sequential processing, the master file, which is used to support transaction processing, is organized sequentially by a record key, such as employee number. The transactions to be processed (the transaction file) are sorted by the record key into the same order as the master file. The master file records

are accessed serially, and each record is examined to determine if there is an updating transaction. After updating, the master record is written onto a new master file. If there is no updating transaction, the record is written without change. The result of the updating process is not only a new, updated master file but also the old master file, which has not been altered. The batch of input transactions is assigned a batch number, and it serves as an audit trail. A reference listing of the transactions in each batch is normally prepared. Source documents can be stored in batch order.

Since all processing is performed periodically, with serial access to master file records, the file order medium can be magnetic tape or disk. The file order batch method of processing is illustrated in figure 9-7 on the following pages.

Both operational and control problems are kept to a minimum in batch entry and file order batch processing. The processing batch reference, batch storage of documents, and batch listings provide a clear and easily followed processing trail for control and audit purposes. Also, the sequential file organization is easy to understand, and the retrieval of file records is straightforward. Error and control procedures are also fairly simple. The problem of backup and recovery in the case of processing failure, loss of a file, and so on, is quite simple because the processing methodology results in the creation of a new file. The old file plus the updating transactions are stored for reconstruction purposes.

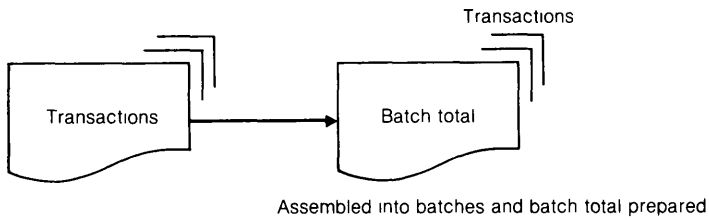
## Summary

In the traditional data processing cycle, documents are used to capture and record source data, in the terminal entry cycle, source data are entered directly.

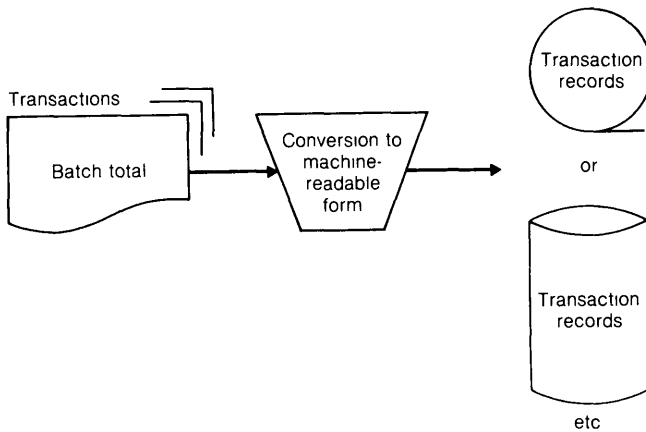
Three major processing methods and eight variations can be defined. The terminal entry/on-line processing method has the control advantage of immediate validation, usually including access to reference data. It has the disadvantages of complexity when transactions are processed as they occur and a lack of precomputed control totals to detect loss or nonprocessing of transactions. The terminal entry/batch processing method has the control advantage of immediate validation for entry plus the advantage of batch controls. Batch entry/batch processing has the advantage of full batch control, which is effective at detecting loss or nonprocessing of transactions and some types of errors in the entry of amounts. The disadvantage of batch entry/batch processing is the delay in detecting, correcting, and re-entering transactions.

**Figure 9-7** File-order batch processing

Step 1: Batching of transactions



Step 2: Conversion of transactions to machine-readable form



Step 3: Input and input validation

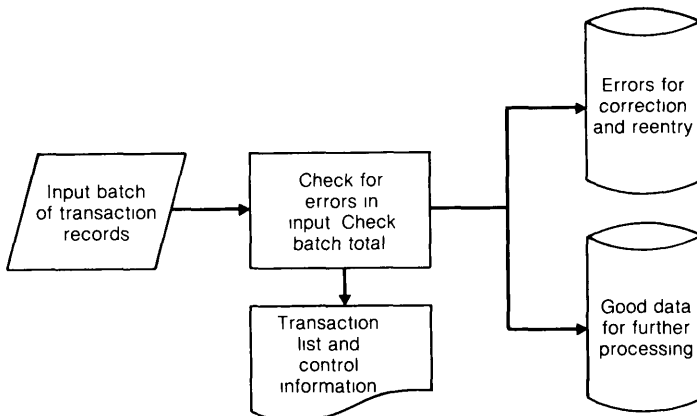
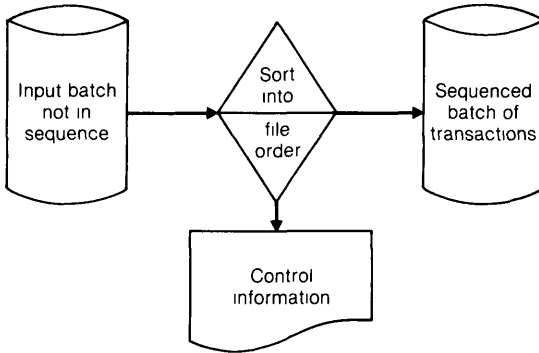


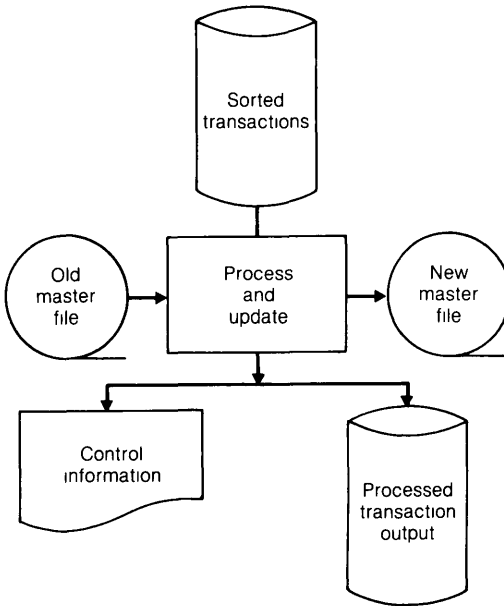


Figure 9-7 (continued)

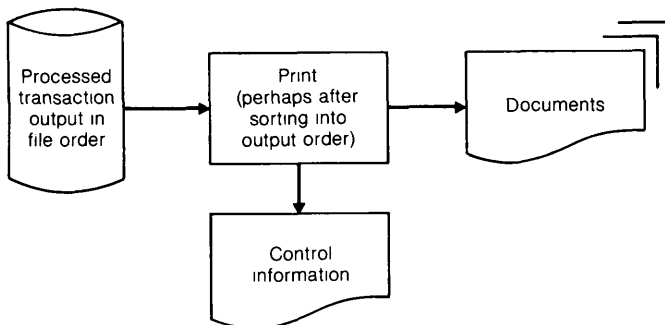
Step 4: Sequencing of transactions



Step 5 Process transactions and update master file



Step 6: Output transaction documents





---

# Application Controls: Authorization, Data Validation, Error Handling, and File Quality Maintenance

# 10

In evaluating internal accounting control, the auditor should identify the key applications that process financial data and produce balances used in financial statements

Depending on the audit approach that is used, an auditor may wish to obtain evidential matter regarding the existence of, and compliance with, suitably designed application controls for some or all of the key applications. An effective approach would be the breakdown of the evaluation into input, processing, and output controls in a way that pays particular attention to authorization, data validation, error handling, file quality maintenance, and audit trail. The controls that are specific to an application should be evaluated in light of the effectiveness of general controls. An effective internal accounting control system does not need to include every possible control procedure. The adequacy of application controls should be measured according to the risk or consequences of errors and irregularities and the cost of the controls.

Authorization, data validation, error handling, and file quality maintenance are crucial to quality control in an application. The controls are specific to the applications, however, they can be evaluated using standard control evaluation procedures.

## Authorization

Each transaction processed by EDP should be authorized by management. There are several methods by which management may authorize transactions, and each of these methods may be implemented in EDP applications.

### Authorization Method

General authorization of transactions commonly performed by a position in an organization, identified by the signature or the authorization code of the person in that position.

### EDP Applications

Transfer of funds using a terminal and authorization code assigned to a bank teller.

(continued)

## 154 Application Controls: Authorization, Validation, and Error Handling

### Authorization Method (cont.)

General authorization if a transaction follows the authorized procedures, uses authorized values, and is within authorized limits

Authorization that follows from the authorization of a prior, related transaction

Specific authorization that requires evidence of authorization or permission, such as a signature or authorization code

### EDP Applications (cont.)

Recording of a sale using values in a table of prices and the approved credit limit

Issuance of shipping documents following the authorized sale of merchandise

Change in application logic prescribed by a change request form issued by a user having authority to request the change

In a manual system, the flow of processing usually involves a set of documents from which the individual who actually processes a transaction can easily verify the signature or initials of the person authorizing the transaction. From these documents an individual can also verify that authorized procedures have been followed before he initiates a related transaction. In a computer application, the authorization at initiation may be demonstrated by (a) source documents with signatures, initials, or evidence of authorized procedures and authorized values or (b) terminal transaction initiation authorization procedures plus records of terminal use.

The terminal entry can be programmed to establish an identification for the person entering the transaction and to check the authority of the person to initiate the transaction. The control over authorized initiation of transactions can be very effective if there are controls for (a) access to the terminal, the application, and the data and (b) changes in stored authorization tables, since a change in the table is the equivalent of a change in authority.

Once the authority to initiate a transaction has been established, the identification of the basis of authorization can be included in the transaction documentation. The authorization may be explicit (from an individual) or implicit (from following authorized procedures). The procedures that establish authorization for transactions may be manual or contained in computer program logic. When procedures that establish authorization are contained in computer programs, a change in the logic is equivalent to a change in the authorization. An example of this is the processing of a sale. The sale is authorized if it is entered for an approved customer and uses authorized prices and procedures. A change in computer logic for entering the sale, checking availability, pricing, and so on, affects the authorization.

Approval is closely related to authorization. Authorization generally occurs before the transaction is initiated; approval occurs during the transaction or after it is completed. Approval indicates that the conditions of authorization have been met. For transactions automatically generated by computer logic, approval procedures ensure that the authorization logic is followed. An example of approval is a purchase order produced when the quantity on hand of an inventory item drops below the order point. Programmed logic plus stored data items are used to calculate the order point, calculate the quantity to order, select the vendor, and produce a purchase order. The approval procedure reviews the purchase order and implicitly reviews the stored logic and data used in producing the purchase order.

## Methods for Validating Data

Data validation procedures screen out incorrect, invalid input data. They are used prior to processing procedures. Validation of input data is sometimes referred to as input editing, but input editing is a broader term, defined in the *American National Dictionary for Information Processing* as procedures "to prepare data for later operation. Editing may include the rearrangement or the addition of data, the deletion of unwanted data, format control, code conversion, and the application of standard processes such as zero suppression." Data validation procedures are used to identify data errors, incomplete or missing data, and inconsistencies among related data items. Data items that do not pass validation must be corrected before further processing is performed. Six major categories of data validation procedures are logic tests, check digits, echo checks, stored data comparison, control totals, and output review.

### LOGIC TESTS

Logic tests are performed by programmed logic. They can be performed by the main computer or by any device that has logic capabilities, such as mini-computers, intelligent terminals, and keydisk units. If the logic capabilities of the device are limited, then logic testing of data may have to be restricted. The logic tests used in a specific case are determined by the characteristics of the data to be checked, the criticality of the data item being tested, and the ingenuity of the analyst and programmer. The following examples illustrate the range of possible tests.

*Valid Classification or Transaction Code* If there are only a limited number of valid classification codes (for instance, for classifying expenses), the code being read may be checked against a table of valid codes to ensure that it is included in the table. There are usually relatively few valid transaction types recorded on a particular file. For example, only a limited number of transaction codes can apply to accounts receivable file updating. The transaction code may be tested against a list of valid transaction codes for the application.

*Valid Item Size, Sign, and Composition* If a code number should contain a specified number of digits, the computer may be programmed to count the digits. If an item should be positive or negative, a test may be made to ensure that the sign is correct. If the item field should contain only numbers or letters, a test may be made to determine that the field contains the proper type of characters.

*Invalid Values* If certain values for a data item are always invalid, a test may be made to check the value of these items. Common examples of invalid values are all blanks or all zeroes.

*Valid Combination of Items.* Combinations of data items may be tested for validity. For example, if a sales representative's code can be associated with only a few territory codes, the program can check for invalid combinations.

*Missing Data Test.* The computer may be programmed to check data fields to ensure that all necessary data fields are present.

*Sequence Test.* If the data records must be entered or processed in a specific order, a test can be made to check the sequence in which data are processed.

156 Application Controls: Authorization, Validation, and Error Handling

*Range and Reasonableness (Limit) Test* Data values usually can be defined as falling within certain limits. For example, hours worked should not be less than zero and should not be more than sixty. (The upper limit may be based upon the experience of a particular firm.) Data may be compared with this limit to ensure that no error, or at least no error exceeding certain pre-established limits, has occurred. For situations in which the logic test for limit and reasonableness has access to historical data, the limit may change for each transaction. The following are examples of variable limits:

- The total amount of a customer's order may be compared with the amount of his average order. If this order exceeds three times the average order, an exception notice may be printed.
- A material receipt that is more than double the economic order quantity established for the particular item may be subject to question.
- An amount on a receiving report may be compared with the amount on the purchase order. If there is too great a difference between the figures, an error may have occurred.
- In a utility billing company, consumption is checked against consumption in prior periods to detect possible errors or trouble in the customer's installation.

**CHECK DIGITS**

A check digit is a redundant digit added to a code or identification number for validation. Check digits may be added to charge account numbers, employee numbers, bank account numbers, and so on. Numbers with check digits are frequently termed "self-checking." The elements necessary to a check digit assignment are the modulus or base, the weights, and the location of the check digit within the self-checking number. The general procedure for assigning a check digit consists of five steps:

<u>Procedure</u>	<u>Example</u>
1 Start with the number without the check digit	57648
2 Multiply each digit by the weight assigned to the digit position	$\begin{array}{r} 5 \quad 7 \quad 6 \quad 4 \quad 8 \\ \times 1 \quad \times 2 \quad \times 1 \quad \times 2 \quad \times 1 \\ \hline 5 \quad 14 \quad 6 \quad 8 \quad 8 \end{array}$
3 Add the digits derived from the calculation in step 2	$5 + 1 + 4 + 6 + 8 + 8 = 32$
4 Subtract this sum from the next higher multiple of the base. The difference is the check digit	$\begin{array}{r} 40 \text{ (base 10)} \\ -32 \\ \hline 8 \text{ (check digit)} \end{array}$
5 Insert the check digit as part of the number (in any position)	576488

The check digit procedures are designed to detect the typical errors encountered in transcribing a number—transcription error (writing the wrong digits) and transposition errors (reversing the position of adjacent digits). The effectiveness of the check digit procedure depends on the modulus and digit weights.

Base 10 and base 11 are most commonly used to develop self-checking identification numbers. Base 11 is more effective at error detection than base 10, however, it generates eleven different check digit values (0–10). If a single check digit is required

in base 11, the value for 10 must be represented by a nonnumeric character, such as A, otherwise, identification numbers with check digits of 10 must be eliminated from the set of available codes

A number of weighting patterns can be used in developing self-checking numbers. The most common are patterns such as 1-2-1-2 and sequences such as 7-6-5-4-3-2-1 or 11-7-5-3-1 (the prime numbers). Although the pattern 1-2-1-2 is common, it has no power to detect errors of double transposition. For instance, the error of writing 57846 instead of the correct number, 57648, would not be detected because both yield the same check digit.

The check digit procedure is very effective and should be included in all new code and identification numbers. Check digits may be added to existing numbers. For example, the bank routing number encoded on checks did not have a check digit, a base 10 check digit has now been added. The social security number does not have a check digit, but some companies now add a check digit suffix for internal processing.

### **ECHO CHECKS**

A validation procedure for input data is to echo the data items back to the person who prepared them or entered them at a terminal. This procedure is especially effective with terminal input.

The echo may provide descriptive output to assist in the review. For example, at a terminal, the operator inserts a customer identification code of 65431, and the terminal echoes the input by the following message:

```
CUSTOMER 65431 LDS CONTRACTORS
```

As an example of echoing in a batch report, an invoicing run to remove all invoice balances of less than a given amount should echo the amount being used as an input item:

```
AMOUNT USED FOR CANCELLATION $1 00
```

### **STORED DATA COMPARISON**

One objective of stored data comparisons is to ensure that the correct record is being processed. For example, an input transaction may include the record identifier and one other item of data from the record. The input record is assumed to be correct only if both items match. For example, a savings and loan company that did not have a check digit on its account number used the combination of account number plus the last recorded passbook balance to ensure that a transaction would not be processed to the wrong account.

Change transactions that alter values (such as pay rates) can require the input for the transaction to include the old value as well as the new value as a check against error. Before the transaction is processed, the old values are compared to validate the change transaction.

Stored data items are also very important in limit tests that depend on the characteristics of the customer or employee, such as average purchase, past consumption, and so on.

### **CONTROL TOTALS**

Control totals can be used in input validation and in control over subsequent processing. First, a control total is computed prior to the start of the process being checked.

## 158 Application Controls: Authorization, Validation, and Error Handling

The control total is then recomputed by the process. The two totals are compared, and any difference between them indicates an error. The error may have occurred in the data processing, in the computation of the control total, or in the procedures for comparing the two totals. A control total is effective in detecting that data were lost in processing or that data were not processed at all. It can also detect records that were added during processing and errors in particular data fields if they are part of the control totals.

Control figures may be financial totals, hash totals, or document or record counts. Financial totals are amounts, such as sales, payroll amounts, and inventory dollar amounts, that are normally added together to provide financial summaries. Hash totals are totals of data fields that are usually not added together. The hash total has meaning only as a control and is not used in any other way in data processing. For example, a hash total may be computed for the inventory item number to be used as input to determine that all inventory items are processed. This control total is compared with the sum of the item numbers obtained during the processing run. A document or record count is a tally of all the documents and records to ensure that all the data have been processed.

### OUTPUT REVIEW

Output review can be used as a validation procedure following processing. The review can be performed by control personnel before output is distributed and by user personnel when output is received. The review is aided if the output contains sufficient data for the reviewer to calculate or estimate critical output values. In addition, values used in the computations that were obtained from files or stored tables may have to be included in the output to provide adequate data for the review. Figure 10-1 illustrates the output data that can aid in the review.

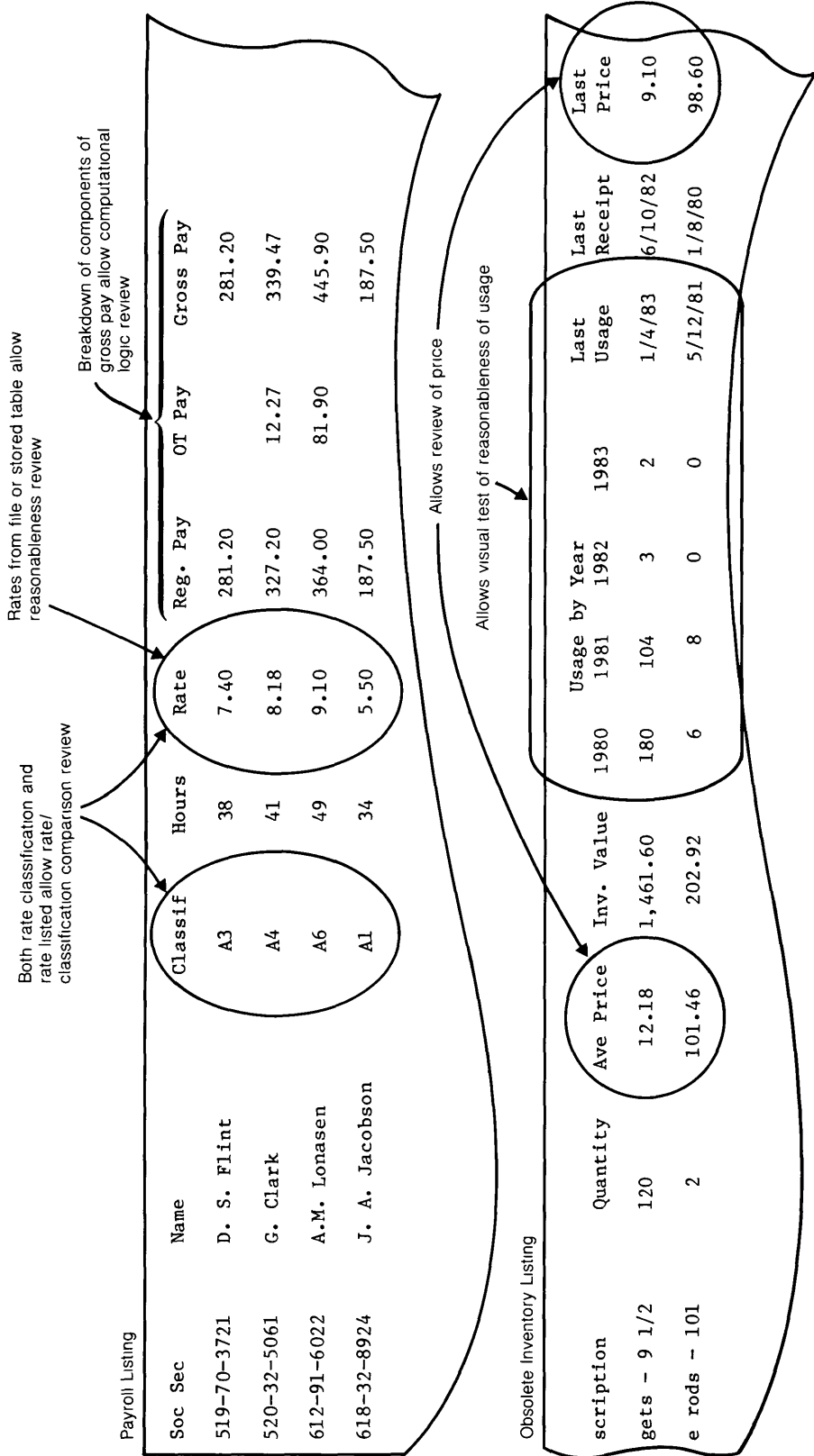
### GENERALIZED INPUT DATA VALIDATION

Data validation tests for errors or irregularities. Since data validation is different for each application, it is part of the application design. However, there are a limited number of validation techniques. Most of the differences in the procedures for data validation result from differences in test values rather than from differences in the testing process. This similarity makes it possible to use a generalized input data validation, especially for logic tests and check digit validation. A generalized data validation routine is used for several applications, each application uses the validation module plus its own validation specifications, which are stored in a table or file.

A good illustration of the use of generalized data validation routines is the use of a common routine for input to batch processing for several applications. Using a common validation routine, input for batch processing can be combined in a single validation run (figure 10-2). Each transaction is given an application identifier and a unique transaction identifier. As application transactions are validated, the valid transactions are written on a pool (a common transaction file) and errors and irregularities are written on an error file. Control totals are maintained for both accepted and rejected transactions for each application. Since each transaction in the pool has an application tag, it is relatively simple to assemble the input batch for periodic batch processing of each application.

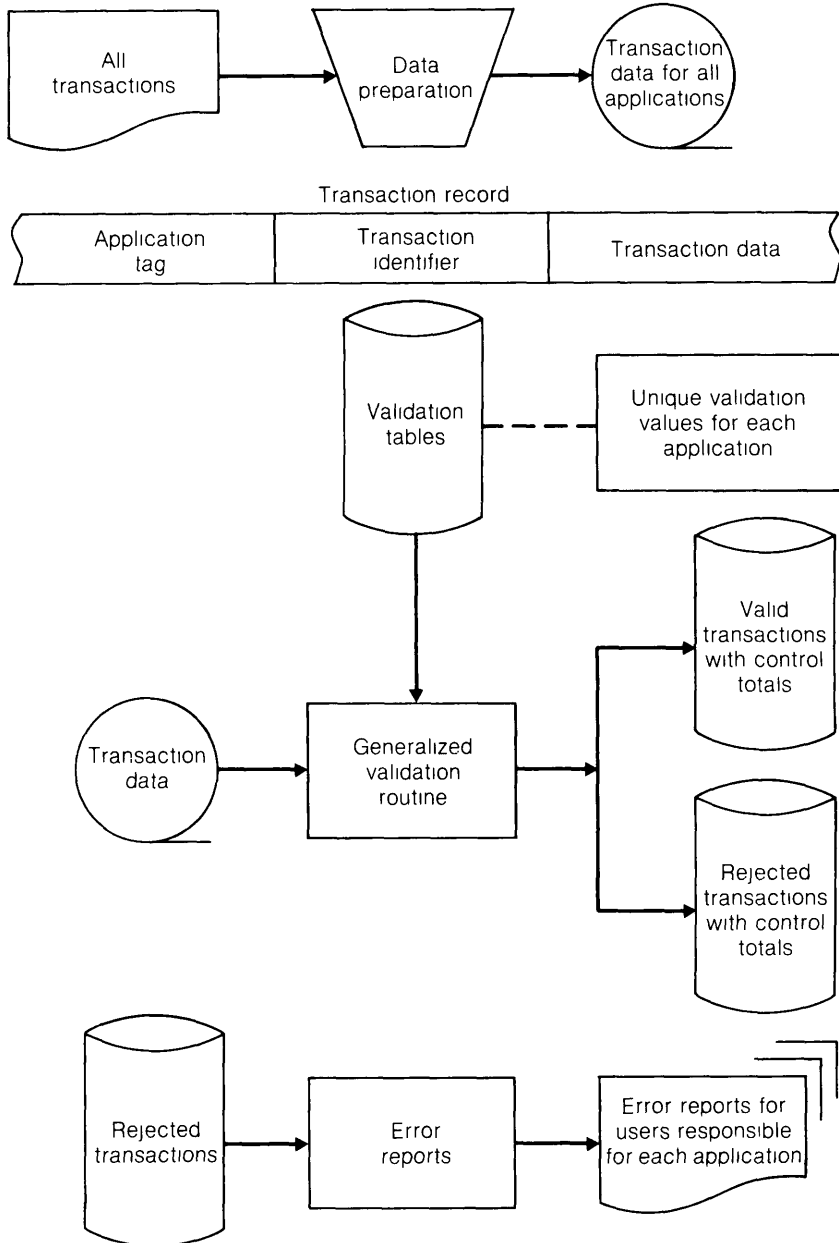
There are significant advantages in this approach. Validation of data can be performed on a continuous basis, even though updating for an application occurs only periodically, for example, weekly. The risk of loss of data is minimized because data



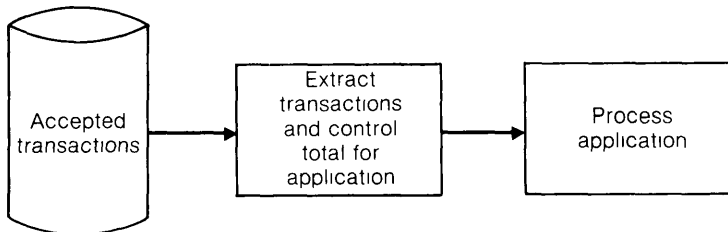


**Figure 10-1** Examples of output data that can aid in output review for reasonableness

**Daily**



**Periodically**



**Figure 10-2** Generalized input data validation

records are validated and written on the pool shortly after they are created. Source documents, for example, do not have to be stored to await processing. A further advantage is the fast feedback of errors provided to users, who often can correct errors quickly when the data are recent because they are familiar with the transactions.

## **Control Over Error Notification, Correction, and Resubmission**

Procedures for control over errors are quite simple in immediate terminal entry and processing, input is validated, and any transaction containing an error is rejected. When data items are batched and processing is delayed, error control becomes more complex.

The major objective of error control is to ensure that all corrections are made properly. In other words, procedures should ensure that all corrections are made, that they are made only once, and that the correction is itself correct.

### **TIMING OF CORRECTIONS**

The delay method and the reject and re-enter method are the two approaches used in processing corrected data items. In the delay method, all processing is suspended until the erroneous transactions are corrected. A preliminary run validates transaction data and identifies errors. The transactions containing errors are reported back to a user who has responsibility for making corrections. The corrected transactions plus previously validated transactions are combined, and the validation and processing procedures are continued. The delay method is used in cases for which it is important to have results for all transactions, for example, payroll processing to produce pay checks and accounts receivable processing to produce billings.

The reject and re-enter method is used when there are frequent processing runs. In this way, the erroneous data may be dropped from the processing run in which it appears, corrected, and re-entered in a later run, without any interruption in processing. When transactions are rejected, control totals for the batch must be adjusted. Situations in which the reject and re-enter method is appropriate include daily processing of accounts receivable transactions (sales and payments on account) and daily processing of inventory transactions.

### **VALIDITY OF CORRECTIONS**

Corrected data should be subjected to the same data validation applied to original transactions. Correction may require a complete replacement of the erroneous transaction or replacement of only the erroneous portion of the transaction data. In either situation, it is a good control practice to revalidate the entire corrected transaction.

### **CONTROL OVER RESUBMISSION**

Data items that contain errors must be carefully controlled to ensure that they are resubmitted (so the transaction is not lost) and that only one correction is made. An input error listing or error report that explains why each item was rejected should be prepared. The report should be returned to the originator of the data or to the person responsible for correction and resubmission. Personnel receiving the error reports should be instructed in the handling of correction and resubmission of data.

The disposition of the erroneous transaction may depend on the type of transaction and other control considerations. The four possible dispositions are (a) to return the transaction to the originator, (b) to hold on a common error suspense file for correc-

## 162 Application Controls: Authorization, Validation, and Error Handling

tions, (c) to process the data with an error flag, or (d) to write a suspense record in an application file

- 1 *Return the transaction to the originator* No copy of the erroneous transaction is maintained in data processing, but the transaction may be noted on an error log. The entire transaction must be resubmitted.
- 2 *Hold on a common error suspense file* The transaction is written on an error suspense file used for a number of applications and held until corrections are submitted. There is no need for the entire transaction to be resubmitted. The correction identifies the transaction and specifies the data items to be replaced. When the correction is received, the transaction is removed from the error suspense file, corrected, and entered into the processing system, starting with data validation.
- 3 *Process data with an error flag* In some instances, the erroneous transaction must be processed. For example, a new master record may need to be established for an account even if there are errors in the data. In such an instance, the record is created, but an error flag in the record is set to the "error" value. When the correction is made, the error flag is reset to the "no error" value.
- 4 *Write a suspense record in a single application file* If a transaction is to be processed against a master file record but cannot be identified with any master record, the record can be written on a suspense file. This method must be modified for those cases in which the master file should total to a control account balance, because placing the account on a suspense record will throw the detail file out of balance with the control account. Examples of this situation are accounts receivable and inventory. In order to use the suspense file to hold unidentified detail records and to keep the master file in balance at the same time, a suspense record may be maintained in the master file. An erroneous detail record is flagged and written on the suspense file, and the amount is added to or subtracted from the suspense ledger record in the master file. When the transaction is corrected, it is processed against the correct account and removed from the suspense record. Suspense records should not be used for balancing unless there are strong controls for follow-up and removal from the suspense file.

The control function in a data processing organization should log all errors that are out for correction. All corrections that are submitted must be checked off. Items not corrected in a suitable time should be followed up. The error log can be maintained manually, but it is easier to maintain by computer. A general procedure for error handling is the circulating error file.

### **CIRCULATING ERROR FILE**

The generalized validation module (described earlier in this chapter) provides the basis for a circulating error file, which is essentially a common error suspense file. Input data records for all applications are processed through a common validation procedure, using the specific criteria for the application to which each transaction belongs. Each transaction is given a computer-generated identification tag when it enters the validation routine. Error transactions are written on an error file, and each transaction has both an application identifier and a transaction number for identification. All error transactions are reported in a standard format to the user area or to the area responsible for error correction.

Data can be removed from the error file only by submission of items to correct the initial data (figure 10-3). If a data item again fails the validation tests, it is rejected, and

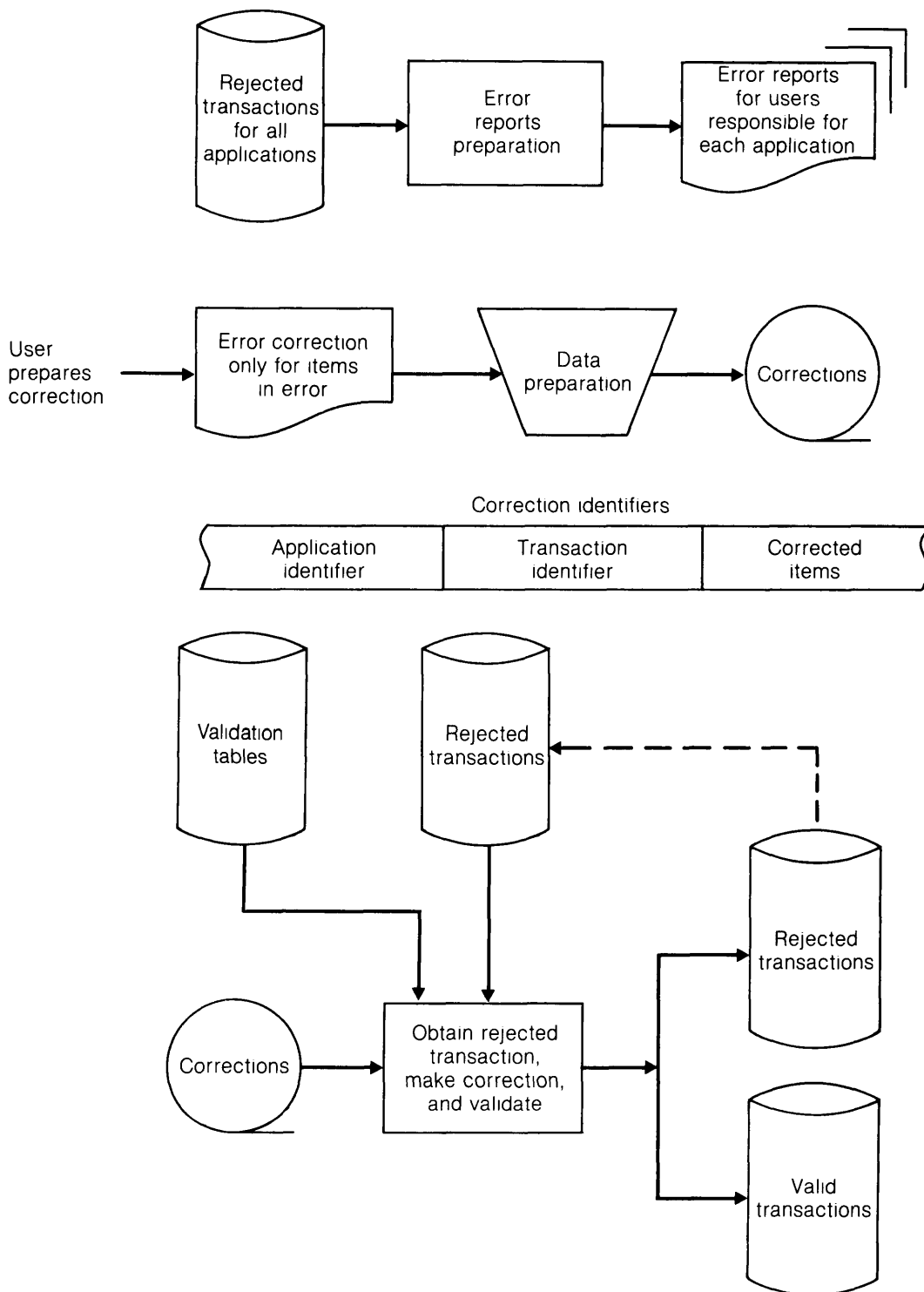


Figure 10-3 Circulating error file

## 164 Application Controls: Authorization, Validation, and Error Handling

the original error remains on the error file. Reminder notices are sent to the respective users for uncorrected data items not removed from the error file. The system continues to print these notices until some positive action is taken to remove the uncorrected items. Control totals for valid data and errors are maintained for each application.

The use of an error file in this manner is extremely beneficial for the control of errors. From an auditing point of view, it provides valuable information about the error performance of a system.

### **File Quality Maintenance Procedures**

Two general methods for regular file quality maintenance for all applications are regular file quality reports and regular file detail reviews.

#### **FILE QUALITY REPORTS**

At regular intervals, all file records may be scanned by a review program, from which a report is prepared for all major users of the file. The logic of the scan is similar to input data validation. Data items are examined using established criteria. Some conditions that should be noted are the absence of data in a field, out-of-range data, invalid codes, and unusually old detail trailer records. The user receives a report on these conditions and a form for submitting corrections.

#### **FILE DETAIL REVIEWS**

The contents of each master file record for major accounting applications should be reviewed periodically, perhaps once a year. The complete record is printed in a form that is easily reviewed. The detail review may be conducted by the user department or by those on whom the record is maintained, for example, employees for employee records or customer account representatives for customer records. The review may be conducted on a cycle basis. The same detail review procedures should be followed for stored tables used by application programs. Examples are tax rate tables and validation criteria tables.

### **Summary**

Application controls are specific to each application, and they are designed to prevent, detect, and correct errors and irregularities. The auditor is interested in the controls over the flow of transactions from data preparation, input or data entry, processing, and output, as well as authorization, data validation, error handling, and file quality maintenance procedures.

Authorization procedures in an application processed by computer may be the same as traditional document preparation procedures or be part of terminal transaction entry. Authorization based on compliance with standard procedures using authorized values can be designed into computer programs.

Data validation is significant in the detection and correction of input data errors and irregularities. The six methods described in this chapter are logic tests, check digits, input echo check, stored data comparison, control totals, and output review. Many installations have made input validation a generalized process, especially for logic tests and check digits. A generalized data validation module performs these tests using the input data record and a set of specifications for the tests stored in a table.

It is vital to establish control over error handling for errors detected during data processing. The error handling procedures should ensure that all errors are corrected and corrections are submitted on a timely basis. Also the procedures should ensure that the same corrections are not submitted more than once. One method for establishing this control is a circulating error file to hold a record of all errors not yet corrected.

File quality depends on original file quality and processing quality. Errors can be introduced in various ways. One control procedure is to scan the file periodically for errors, producing a file quality report. Another control procedure is to print the file records for user review.





---

# Application Controls: Data Preparation, Input, Processing, and Output

# 11

Application controls are used to prevent, detect, and correct errors and irregularities as transactions flow through the application procedures. This chapter presents the application controls in terms of data preparation, input or direct entry, processing, and output. Output distribution control is included as part of output controls, since output must be distributed only to authorized users. Processing controls include both program logic and operating procedures. The chapter describes the potential sources of errors in each of these processing activities and explains the controls that prevent or detect the errors.

## Potential Sources of Errors in Data Preparation, Input, or Entry

The input data provided for the processing activities for an application may be in error for any number of reasons. The most common sources of errors are listed below, along with the types of controls typically used to prevent or detect the errors.

### Source of Error

Failure to record a transaction

Incorrect or incomplete recording of data on a transaction document

Duplicate recording or entry of transaction data

### Controls to Prevent or Detect Error

Procedures to ensure the recording of a transaction

Prenumbered documents and document number reconciliation

Control totals from related activities

Transaction anticipation control

Data review procedure

Data validation

Control totals

Procedures to enforce prompt data recording or data entry at the time of the transaction

Document cancellation

Data review

Data validation

Control or proof figure

(continued)

Source of Error (cont )

Loss of the transaction document in handling

Errors in converting a document to machine-readable input

Incorrect entry of transaction data at a terminal

Controls to Prevent or Detect Error (cont )Movement log  
Documented numbering  
Control totalsVerification  
Data validation  
Control totalsData validation  
Expanded echo of data being entered  
Control totals**Control Over Batch Data Preparation**

Batch data preparation, called more simply "data preparation," refers to the procedures that assemble batches of documents, establish batch controls, and transcribe the data to a machine-readable medium. Data preparation is performed on data items that have been recorded on documents to produce machine-readable records, which can be used for a batch processing input run.

Control to prevent or detect errors and irregularities begins with the design of the document itself and procedures to record, edit, and review data. The documents containing the data are assembled in batches, and control figures are established. Transmittal and routing controls are applied as the batch of documents moves through data preparation. The data items on the documents are keyed into a machine-readable medium. Often, they are also verified at the same time. The input batch is processed to validate the data and to prepare it for further processing.

**PROCEDURAL CONTROLS, EDITING, AND DATA REVIEW**

Well-designed documents and certain standard practices impose procedural controls on the creation of data. For example, if a part number is to be written on a document, the spaces for the part number can be marked so that a number containing fewer or more than the required characters will be detected by the person preparing the document. When terminal devices are used, templates over the keys, identification cards, screen formats, and other devices serve to reduce input errors.

Editing and review procedures may be used to examine input data items before they are converted to machine-readable form. These procedures may be part of a manual operation to add information, such as code values, to transactions, or they may be separate steps to check transactions. Sight verification of input data can be used when a visual display terminal is available as the data preparation device.

**PREPARING BATCHES OF TRANSACTION DOCUMENTS**

Batch processing requires the grouping of related documents—either original documents or turnaround documents prepared by prior processing. The size of the batches will vary, as will the types of control totals that are used.

The size of a batch is limited by the number of transactions that occurred during a period and by the time required to check a batch if an error is detected. Control totals can indicate only that the entire batch is in error; personnel must then search for the erroneous transactions. A reasonably small batch size facilitates error locating procedures, but it means that more batches must be created, logged, listed, and totaled. A reasonable batch contains anywhere from fifty to two hundred documents.

After a batch of documents has been transcribed, it may be stored for reference. The batch is cancelled by a stamp, processing mark, or perforation to prevent its resubmission for processing.

### TRANSMITTAL CONTROLS AND ROUTE SLIPS

Data in transit in an organization may be lost or diverted from the proper processing channels. To ensure proper identification of data as they move through the company, and more especially as they move through the data processing steps, some form of identification should be used. For instance, as batches of data enter the data processing center, they may be logged on a listing showing the date they were received. As each batch passes a data processing station, it may be registered to indicate that it has been processed. In addition, the batch itself usually carries a route slip that indicates its processing path and supplies a record of the processing already performed.

### KEY ENTRY TO MACHINE-READABLE FORM

Key entry of data is performed via a keydisk or keypunch device. Batches of documents are transmitted to the key entry supervisor, who logs in each batch. The supervisor then assigns the batch to an operator, who selects the proper key entry format control. If a sophisticated key entry device is used, the operator may enter the control total before keying the data. The device will then accumulate the same control figure for the data items being keyed and compare the two totals.

If the source data are arranged in a confusing manner or if the source documents are written illegibly, the operator may key in erroneous data. The operator may interpret the data correctly but strike the wrong key, thus entering erroneous data; an experienced operator averages one error for every 1,500 to 4,500 keystrokes. In more modern equipment, the keystrokes are stored in a buffer and are recorded only when the operator strikes a release key, so that an operator who strikes the wrong key can correct the error before the keystrokes are recorded.

Keydisk equipment provides visual display of data for immediate verification, which further reduces error rates. For some equipment, additional logic capabilities allow an operator to perform extensive validation procedures during data entry.

Methods of data entry are listed below, along with notes on the entry devices' validation capabilities.

<u>Device</u>	<u>Data Storage</u>	<u>Validation Capabilities</u>
Keydisk	Disk with transfer to magnetic tape	<ul style="list-style-type: none"> <li>● Valid classification or transaction codes</li> <li>● Valid item size, sign, and composition</li> <li>● Range tests</li> <li>● Check digits</li> <li>● Balancing of batch totals with control totals</li> </ul>
Keypunch	Cards	<ul style="list-style-type: none"> <li>● Check digits (on some models)</li> <li>● Other keydisk capabilities (on some newer models only)</li> </ul>
Key diskette	Diskettes	<ul style="list-style-type: none"> <li>● Check digits</li> <li>● Control totals</li> </ul>

(Continued)

## 170 Application Controls: Data Preparation, Input, Processing, and Output

<u>Device (cont )</u>	<u>Data Storage (cont.)</u>	<u>Validation Capabilities (cont )</u>
On-line CRT	Disk	<ul style="list-style-type: none"><li>● Valid classification or transaction codes</li><li>● Valid item size, sign, and composition</li><li>● Check digits</li><li>● Balancing of batch totals with control figures</li></ul>

### KEY VERIFICATION

In key verification, a second operator rekeys part or all of each document in the batch. The keying device compares the two sets of data and indicates any discrepancies. For example, in the verification of cards, a notch is placed in a card over the column in which there is a discrepancy. If there are no discrepancies, the notch is placed at the right-hand edge of the card. In keypunch entry, corrections are made by a separate operator and reverified, in keydisk, key diskette, and on-line CRT entry, the verifying operator usually makes the corrections.

Verification is a duplicate operation that doubles the cost of data conversion. Therefore, various methods are used to reduce the amount of verification. One method is to verify only part of the data. Some data fields are not critical and an error in such a field will not affect further processing, examples are descriptive fields containing vendor name, part description, and so on.

Another way to reduce verification, used primarily with statistical data, is to verify the data only if the key error rate is above an acceptable level. A sample of each operator's work is checked. If the error rate is acceptable, no verification is made, if it is unacceptable, complete verification is required.

### MACHINE-READABLE DATA INPUT PREPARATION

Machine-readable data can consist of specially prepared documents, data recorded on a magnetic medium by a transaction or data collection device, or turnaround documents. The major technologies used for machine-readable documents are optical character recognition (OCR) and magnetic ink character recognition (MICR). The readers are designed to interpret documents containing only a few lines of data or an entire page of information. Both OCR and MICR use a special device to read batches of documents and to store the data on a medium such as magnetic tape.

OCR is able to read a variety of codings, such as marks, bars, printed characters (in stylized fonts such as OCR-A or OCR-B), and hand-printed characters. Error control at the preparation of documents using OCR characters includes visual verification of the recorded data. Error control of reading by OCR is limited to check digits on the identification fields. Logic checks and matching of control totals are performed during the input run. Unreadable documents or characters are handled in one of two ways. The document may be rejected, and a separate input is prepared by keying the data. Alternatively, unreadable characters may be placed on a visual display unit, so that the operator can examine the document and key in the correct characters.

MICR is used extensively by the banking industry. Error control of reading by MICR includes check digits and batch totals. MICR coding is also used to sort documents. For example, MICR-encoded checks and deposit slips are sorted in customer number order so they can be placed in customer statements. Unreadable documents are rejected, corrected, and re-entered.

There are several means of collecting data at the point of transaction, including point-of-sale devices and data collection devices. Some of the devices store data on magnetic tape cartridges, others store data on magnetic disks or tape. Some of the devices can perform logic checks, check digit verification, and control totals.

Turnaround documents (used extensively in book club and billing applications) are documents that have been returned by customers to complete a transaction. They are typically machine-readable, either punched or coded for optical character recognition. The customers' account number and the amount due are already encoded. Data are added only if the amount paid by the customer is different from the total amount due. If necessary, the actual amount paid is added by keypunching, optical character printing, or handprinting. Controls over the reading of turnaround documents are the same as those used for the basic media being used.

## Control of Input or Direct Entry

Three major methods are used for the input of data into processing:

1. Periodic preparation of batched transaction documents and periodic processing of the batched transactions
2. Immediate terminal entry of transactions, which are stored for periodic processing in batches
3. Direct terminal entry and immediate processing of transactions

The timing of data validation differs for the three methods. Direct terminal entry has immediate data validation. Entry at the terminal for subsequent processing may have immediate data validation, depending on the system design. In general, immediate data validation with immediate error feedback is preferred over delayed validation with delayed error feedback.

### BATCH INPUT

The batch input method incorporates a number of control features. Prior to data preparation, the source documents are assembled into batches, and control totals are established for them. A batch ticket is attached to the input batch with the control totals on it. After data preparation, the input consists of the batch control figures and the individual transaction records. A listing of the batch may be prepared prior to input validation for use as a reference and proof listing. This list may also contain control totals. The input validation run produces output that contains all the records accepted and an error listing containing all the rejected records. Control figures are then reviewed and reconciled by the control personnel. If the transactions have an order or sequence, they are sequenced just before or just after input validation, and the batch is checked for missing and duplicate transactions.

### TERMINAL ENTRY AND BATCH PROCESSING

The batch controls for this input method are the same as those for the batch preparation and input method. The major control differences relate to the use of a terminal, which allows immediate data validation and immediate error feedback and corrections. Since the transactions in this method are stored for subsequent processing, they may be grouped into physical batches prior to processing. Control totals are used for each batch and for the sum of all batches being assembled. The terminal entry batches and batch totals are entered into processing in the same manner as regular batch process-

ing Since some data validation has already been performed, further validation may emphasize sequence checks, checks for complete sets of transactions, and duplicate transactions checks, which cannot be performed on individual transactions

The need to control access through the transaction terminal depends on the consequences of unauthorized access. If the batch listings and control totals provide adequate control, then normal transactions may be allowed with little control over terminal access. On the other hand, for critical transactions, such as changing master file data or altering internal tables, access to the terminal and entry of certain transactions should be controlled.

### **IMMEDIATE TERMINAL ENTRY AND IMMEDIATE PROCESSING OF INDIVIDUAL TRANSACTIONS**

Immediate terminal entry followed by immediate processing provides significant control features. Data validation can include all validation procedures except sequence and complete set of transactions. The access procedures can test for authority to enter a transaction or receive output. The design of the input procedures can reduce errors and provide for recovery and restart when entry procedures fail. Conventional batch totals, however, are not normally available before entry or before processing. Post processing batch or logical batches may be used to establish control totals and reference lists of processed transactions.

The on-line validation procedures not only provide immediate feedback and correction but also assist in visual verification by providing supplementary data. For example, the operator may enter an account number. The name of the account is then obtained from the master file and displayed on the terminal screen. The operator can then check the name against his records to see if it is correct. Displays of before and after values of a changed data item can also provide visual confirmation of changes that have been made.

The access procedures for input determine a person's authority to enter a transaction type from a specific terminal. In addition, there should be a clear distinction between the functions and authority of supervisory personnel terminals and terminals designated for application transactions.

Data entry may be guided and prompted by a computer program. Such prompting can be an effective approach for an operator who is not yet familiar with the processing procedures. Expert operators, though, will not want to follow these step-by-step instructions, therefore, the software should have an expert mode that bypasses many of the simple entry instructions required for trainees.

There should be clear user instructions on how to handle recovery from any error. The application program should clarify status and aid in the recovery process. For example, if a transaction is entered and the terminal is disconnected because of line failure, the entry procedures and computer-generated messages should clarify whether the last transaction was received and processed.

When a transaction is entered, it is written in a transaction log, on tape or disk. This log aids in backup and recovery. It also allows the preparation of logical batches of listings for transactions that have been processed. These provide control totals and reference lists. For example, a number of bank tellers are connected on-line to a central computer. Different tellers use the same terminal at different times throughout the day. However, on the basis of a teller identification code, the program prepares a control total for deposits and withdrawals entered by each teller. This control total is balanced with the actual daily money proof submitted by each teller.

The control procedures for on-line entry generally require that each transaction be uniquely identified. The most common identification procedure is to have the computer program assign a number to each transaction when it is entered at the terminal.

## Sources of Processing Errors

Application processing controls are designed to ensure that the correct data have been used in processing and that processing is correct and complete. Although processing procedures operate on data records that have been validated by input data procedures, processing logic should not assume that input records are correct. Individual data items may appear to be correct, but when the items are combined in processing they may yield incorrect results. Processing steps should include additional checks to detect not only errors in processing but also errors that may have passed through input validation.

In addition to the possibility of unauthorized or incorrect data that may not have been detected by input validation procedures, processing errors may be due to the following:

- Use of an incorrect version of the program
- Use of a wrong file or record in processing
- Use of an incorrect value in internal tables
- Wrong default value
- Input of incorrect program parameters
- Precision and rounding errors
- Incorrect or incomplete processing logic

### USE OF AN INCORRECT VERSION OF THE PROGRAM

Programs are continually corrected, changed, and improved. Each change creates a new, updated version. For backup, recovery, and similar purposes, it may be desirable to retain an earlier version of the program, thus, it becomes possible for an operator to use an incorrect version of the program in processing. To guard against such errors, a facility may use general library controls and application printouts that identify the version of the program being used.

### USE OF A WRONG FILE OR RECORD IN PROCESSING

This processing error does not usually involve use of a different application file, which would cause a system error because the record formats would not match the program description. Rather, it is more likely to involve the use of the wrong generation of records, for example, use of an out-of-date master or transaction file instead of the current one. This error should be prevented by general controls over the file library, but the program itself should check the file labels. In some cases, this verification may require the operator to enter some comparison data (dates, generation number, and so on) or to review and validate data on the files being used. In other cases, the program and files may have sufficient internal data for an adequate check.

The use of wrong records in processing is not a common problem because it is relatively easy to test for the presence of the correct records. However, it can occur if

## 174 Application Controls: Data Preparation, Input, Processing, and Output

no test is made. A program should not assume that the next record is the correct one, a specific test of record identification should always be used to check that the correct record is being processed.

### **INCORRECT VALUE IN INTERNAL TABLES**

Incorrect table values should be detected by input validation procedures, but, since table values are often crucial to correct processing, further controls are needed. One of the two common control procedures is to produce a printout of the entire table as part of the control output for the application. The printout should be reviewed by control personnel or by application area reviewers. The other common procedure is to print the table value used in a computation as a part of the normal listing. For example, if an employee's pay rate comes from a table of pay rates, the pertinent rate is printed as part of the payroll output for the employee.

### **WRONG DEFAULT VALUE**

Many programs are written to simplify their use by automatically selecting a default option if no specific option is defined as part of the input. For example, a depreciation computation may use one of several methods. If straight line is normally used, the program may assume straight line if no alternative method is specified. Such default selection is very useful, but it may lead to errors if the user neglects to select an option. All such default options should be clearly identified in the output for use in visual inspection and validation.

### **INPUT OF INCORRECT PROGRAM PARAMETERS**

This error can occur in applications for which the operator must enter data via a terminal. When the operator enters the data, special care should be taken to validate the input and to provide descriptive messages that confirm the meaning of data being entered. For example, if the operator is to enter a code of 1 for a regular weekly payroll output and a 2 for a special quarterly output, the terminal should respond with a descriptive message, such as one of the following.

```
CODE 1 REGULAR WEEKLY PAYROLL OUTPUT FOR WEEK ENDED 7/18/81
CODE 2 WEEKLY PAYROLL FOR 9/26/81 PLUS QUARTERLY REPORT FOR
      QUARTER 3
```

Ideally, jobs should be designed to reduce or eliminate the need for operator entry of program parameters.

### **PRECISION AND ROUNDING ERRORS**

Precision and rounding errors do not occur frequently in data processing, they are more likely to occur in computations using complex formulas or algorithms. In certain situations, however, incorrect or inconsistent data processing results may result from these errors.

Precision refers to the number of significant digits available for holding the result of computation. The computer hardware establishes the maximum precision, but a lower precision can be defined for data items that store results. For example, the result of a computation may be stored in a data item having two storage positions beyond the



decimal point, as in *COBOL PICTURE S9999V99*. In this case, only two digits beyond the decimal point will be saved, the remaining digits will be dropped, or truncated. A computed value of 375 76843 will be stored as 375 76

The stored value would be 375 77, though, if the computer rounds the last digit rather than just truncate the excess digits. Thus, the following instructions may yield different results

```
MULTIPLY SAVINGS BY INTEREST-RATE GIVING INTEREST-EARNED
MULTIPLY SAVINGS BY INTEREST-RATE GIVING INTEREST-EARNED ROUNDED
```

Assume, for example, that the result of computation is 36 4762 and that the computer stores values containing two digits beyond the decimal point. The first instruction will produce a value of 36 47, the second a value of 36 48

Some programming languages are designed to automatically round numbers before dropping digits, in others rounding must be specifically programmed. In COBOL the instruction *ROUNDED* must be used

The same group of numbers may be added together with slightly different results, depending on whether (a) the numbers are rounded and then added or (b) the numbers are added in their original form, and the sum is then rounded. For example, a savings plan pays quarterly interest at the rate of 01375 on the first \$100 and 01475 on the remainder. A depositor with \$200 in savings will receive \$2 86 in interest if the following COBOL instructions are used

```
COMPUTE FIRST-SAVINGS-INTEREST ROUNDED = FIRST-SAVINGS * 01375
COMPUTE REMAINING-SAVINGS-INTEREST ROUNDED = (SAVINGS - FIRST-
SAVINGS) * 01475
ADD-FIRST-SAVINGS-INTEREST, REMAINING-SAVINGS-INTEREST GIVING
TOTAL-INTEREST
```

However, the same depositor will receive \$2 85 in interest if the following instruction is used

```
COMPUTE TOTAL-INTEREST ROUNDED = FIRST-SAVINGS * 01375 + (SAVINGS -
FIRST-SAVINGS) * 01475
```

### **INCORRECT OR INCOMPLETE PROCESSING LOGIC**

Programs should be tested thoroughly before they are used, and retested after each program change, in order to detect logic errors. Some programs, however, may branch into so many paths that complete testing is difficult. Since the major processing paths are usually tested, the errors frequently arise in unusual combinations of conditions.

Computer programs typically have sets of instructions that are repeated for each transaction or part of a transaction. Even if the instructions are correct, they will not be executed properly if certain initial conditions are not met, such as cumulative sums equal to zero or program switches set to initial conditions.

The termination of processing for an application (the end of the job) usually requires some last-time-through logic. If this logic is incorrect, all records but the last one will be correct, and, thus, the error may not be noticed.

In order to prevent logic errors and data errors because of unusual combinations, the program should contain internal checks for processing accuracy and completeness.

## Programmed Processing Error Controls

The error controls that can be programmed are only limited by the ingenuity of the designer and programmer and by the cost of programming and executing those controls. The cost includes, for example, any manual analysis, comparison, and follow-up required by the error control procedure. Some useful controls are as follows:

- Limit and reasonableness test (range test)
- Sequence test
- Explicit class identification
- Crossfooting test
- Control totals
- Process tracing data
- Error correction screening

### LIMIT AND REASONABLENESS TEST (Range Test)

Just as in input data validation, processing errors may be detected by program instructions that test the reasonableness of the results by comparing them with either predetermined or flexible limits. The net pay in a payroll application may be checked against an upper limit representing the reasonable maximum pay. Any paycheck exceeding this is probably in error. In a billing operation for a relatively homogeneous product, such as steel bars or plates, the weight of the shipment may be divided into the billing in order to develop a price per pound. If the price per pound differs from a predefined average by more than a predetermined percentage, a message should be written for subsequent follow-up to determine if the billing is in error.

### SEQUENCE TEST

If data items should be processed in a defined sequence, and if transactions are given identification numbers in the proper sequence, then the set of transactions can be tested for sequence and for missing or duplicate items. For example, records of issuance for stock certificates may test for missing numbers in the sequence.

### EXPLICIT CLASS IDENTIFICATION

In programming, there are many situations in which a transaction fits into one of several classes for processing. If, for example, there are three classes of transactions, a program can test for classes one and two. If a transaction does not fit either of these, program logic can assume that it belongs to class three. However, control considerations dictate that the class test should be made until an explicit match is found and that no assumptions should be made for unidentified transactions.

### CROSSFOOTING TEST

Frequently, it is possible to check computer data processing by a method similar to the manual method of crossfooting. Individual items are totaled independently, and a

crossfooting total is developed. In a payroll application, for example, the totals are developed for gross pay, for each of the deduction items, and for net pay. The total for net pay is also obtained independently, by taking the total for gross pay and deducting the totals for each deduction. If this crossfooting result does not correspond to the original result for net pay, there is an error in the program or processing.

### **CONTROL TOTALS**

Control totals, developed in the same manner as the input control totals, can be used for testing the correctness and completeness of data processing. For example, the number of items to be invoiced on a billing run can be used as a control total and compared with the number of items billed on the invoices. Where relevant, control figures developed during processing should be in a form that can be compared with related input control totals. If input controls are on gross debits and credits, then the program should also develop gross debit and credit totals. Control figures should normally include the number of records processed to detect nonprocessing of input records.

### **PROCESS TRACING DATA**

Tracing data can be used for visual inspection to evaluate whether or not processing is correct. Such data may be part of the regular output that is reviewed by the application area, such as printout of the pay rate as part of the output for each individual, or it may be a special listing, such as all invoices past 120 days. Tracing data for changes in critical data items may include before and after images of data field content.

### **ERROR CORRECTION SCREENING**

In some applications, it may be possible to enter valid correction transactions but still come out with an incorrect balance because transactions in process were not considered in making the correction. For example, an inventory balance is adjusted to zero when there are no items in stock, even though the records show a balance on hand. The discrepancy may be due to an unprocessed transaction that is being held at some point, for example, an order held in the warehouse for approval of a substitution. When this transaction is released and processed, the inventory may end up with a negative balance for the item. Another example is removal of interest charged on a disputed customer balance, followed by the improper addition of interest for the current period. In such cases, it is desirable to make program tests to prevent the processing of delayed transactions (or at least to make sure they are noted for review) and to block subsequent transactions that are no longer valid.

Methods for preventing the errors cited in the two examples illustrate the type of error prevention that can be programmed. In the case of the inventory adjustment, a temporary transaction date filter may be stored in the adjusted record. Any transaction dated earlier than the adjustment filter date is rejected and printed out for manual review. The addition of interest subsequent to a removal of interest charged can be blocked by the use of a logic switch stored in the customer record. For example, a value of 1 might mean normal processing, 2 might mean no interest charge this month only, and 3 might mean no interest charge until account status is resolved.

## Processing Errors and Operating Controls

The program design and operating procedures should provide detection and correction of processing errors associated with operations

### Error Condition

Interruption of the application processing run  
 Destruction during processing of part or all of a file  
 Use of the wrong file

Database data errors  
 Errors in applications having several sets of programs in separate runs

### Control Procedure

Breakpoint data and restart procedures  
 Backup of data and recovery procedures  
 Internal file labeling  
 External labeling  
 Librarian  
 Database administration function  
 Run-to-run controls

### **BREAKPOINTS IN PROCESSING**

A batch processing run may require several hours of processing time. If there is a processing error, the run may have to be repeated from the beginning unless rerun breakpoints have been established. When a rerun breakpoint is reached in a program, all data items necessary to restart processing from that point are recorded on a rerun file. Typical data items required for restarting are record counts, control figures, and cumulative sums.

Rerun provisions are established for each application program. For example, in COBOL an *I-O-Control* paragraph specifies the points at which breakpoints are to be established and the file on which the rerun data is to be recorded.

### **BACKUP PROVISIONS**

The operations for each application should include backup and recovery procedures in case part or all of a master file is destroyed during processing. For sequential files on magnetic tape, the file must be rewritten when it is updated, which automatically provides the father-grandfather files for backup. Since direct access files are updated in place, a backup copy of the file must be made prior to the processing of current transactions. When on-line updating is employed, various files are copied at the beginning of each day. In some cases, backup copying may take place during the day.

### **FILE USAGE CONTROLS**

The procedural controls for file usage include the use of external labels to identify files and the use of a librarian to control access to files.

The use of an internal file label is a control that may require specific programming in the application system or in its job control statements. A file label is a record identifying a file. On a magnetic tape it is the first record, for disk files, the location of the label may depend on the labeling procedure that is used. The information on a header label is often specified by the programming system. Typical data include the name of the file, the creation date, the identification number, and the reel number. There are computer program instructions to write a label on a file being created and to read (and test) a label on a file being read. For example, the COBOL instruction to OPEN a file performs the file label reading and writing.

## **DATABASE ACCESS CONTROLS**

The use of a database assigns responsibility for data to the database administrator who controls the database management software. The database management software provides all data handling functions for properly authorized application software. The authorization is selective, it may be only to read file data or both to read and to write. Data validation may be provided in part by the data management system and in part by the application program. (See chapter 7.)

## **RUN-TO-RUN CONTROLS**

The operational control procedures for an application, when possible, should use control totals to ensure that no transaction records are lost as the records move from data recording to data preparation to processing. Within processing, control totals are important mechanisms to detect loss or nonprocessing of transaction records due to a processing error or a program logic error.

Processing for an application is frequently subdivided into separate tasks called "runs." This is especially true in batch processing. For example, an application for processing the payroll may be divided into the following tasks:

- Validate input data
- Sort records into payroll master file order
- Update master file and prepare transactions
- Sort transactions into name order within a department
- Print the payroll journal in name order within a department
- Print checks

As the processing from one run transfers data to the next run, it is important to make sure that all records are transferred. The use of run-to-run control totals ensures the completeness of processing and allows the identification of the source of an error. Each run produces control totals that are output for review and input to the next run.

Control totals are not useful unless the control figures of one process or run are compared with control totals developed before the process or run. Therefore, procedures for an application should include control figure comparisons. In simple applications, the data processing control personnel may manually compare the computer-generated control total with the prior control total. In more complex applications, the prior control total is part of the input to the processing run, and the computer compares the totals. Results of the comparison are shown as printed output and as a display on the operator terminal, and the computer-generated control output is reviewed by appropriate personnel.

## **Control Over Output**

### **DISTRIBUTION CONTROL**

Output distribution should be controlled to ensure that only properly authorized personnel receive reports or other output. The operator documentation (the run book) specifies the number of output copies to be prepared by the computer. This may require using multiple-part computer paper, repeating the output, using duplicating printers, or using a separate duplicating process. Output copies are distributed according to the instructions in the current version of the run documentation, and someone in data

## 180 Application Controls: Data Preparation, Input, Processing, and Output

processing should be assigned to control the distribution. This involves reviewing the output, logging the distribution, and preparing and attaching transmittal or release forms. One form of output distribution log is a report distribution sheet used to record the disposition of all copies of the output. A transmittal form identifies such information as the individual department and mail station that will receive each copy. As an added control, a confidential report release form may be used for sensitive output; the control requires the recipient's signature acknowledging receipt of the output. This procedure serves to remind the recipient of the confidential nature of the material.

### OUTPUT REVIEWS

The person responsible for processing controls within the data processing installation should check output before distribution for completeness, for the number of copies, and for the agreement of control totals. If feasible, this person should also cross check with output from related programs. For example, this individual should check reductions of inventory in an inventory control program against cost of sales quantities in a sales analysis program. The review prior to distribution includes scanning of the output for obvious errors, such as lines of meaningless characters or missing fields.

The user department or area responsible for the function related to the output should perform simple error detection and control procedures before accepting the output as valid and correct. The following are examples of possible tests by the recipients:

<u>Output Test</u>	<u>Description</u>
Visual scan	Scan output for reasonableness and completeness.
Test against independently maintained control totals	Compare report results with controls established when input data were transmitted for processing. Compare the ledger balance with output if it is a relevant control total.
Comparison with approximations	Compare actual totals with approximate totals computed by the user.
Comparison with physical counts	Compare physical counts, such as cash or inventory, with output.

Since a small number of control comparisons are useful for any set of output, user documentation for an application may include a short output review checklist for user personnel.

### Summary

The controls that are specific to computer processing applications apply to the flow of processing activities—data preparation, input or direct entry, processing, and output.

The controls over data preparation apply to batch processing. These controls cover document preparation, batching of transactions, transmittal, and verification.

Methods of control over direct entry or data input vary depending on the processing method. Authorization and data validation methods described in chapter 10 and batch control totals are used.

Control over application processing includes programmed processing error controls and programmed and manual controls for operating errors. The programmed processing error controls include limit tests, sequence tests, explicit class identification, crossfooting, control figures, process tracing data, and error correction screening. The control procedures for operating errors are breakpoints, backup, file usage controls, and run-to-run control figure comparisons.

Control over output includes distribution controls, output reviews by control personnel to detect incorrect output before it is distributed, and output reviews and tests by user personnel.





---

# The Audit Trail in Computer Data Processing

# 12

All data processing applications require an adequate audit trail. An inadequate audit trail will lead to internal operational and control problems and to difficulties in making an internal or independent audit. For those applications that have significance for taxation, an inadequate audit trail is in violation of governmental requirements. This chapter will explain the concept of an audit trail and will describe various methods for maintaining an adequate audit trail.

## Definition of an Audit Trail

An audit trail is a set of processing references, data, or logic documentation that enables an investigator to trace the processing of a transaction from its source to inclusion in the accounting records or to trace any amount included in the financial records back to the source or origin of the amount. In other words, the audit trail must allow tracing in both directions. The following list shows the capabilities of audit trail tracing, which is diagrammed in figure 12-1.

1. Tracing of transaction processing from the source or initiation of a transaction through data processing procedures to the results of the transaction
  - To transaction documents
  - To the triggering of other transactions
  - To the addition or deletion of records and changes in items within the records
  - To changes in tables used in processing of transactions
  - To ledger balances and other totals that summarize transactions
2. Tracing from any processing result back to the transaction or transactions that caused the result or to the set of transactions that make up the result
  - From the transaction document back to the persons who initiated, authorized, approved, and prepared the transaction
  - From the triggered transaction back to the transaction that triggered it
  - From the addition or deletion of a record, change in a data item within a record, or changes in reference tables back to the transaction that caused the change

**Figure 12-1** Audit trail tracing

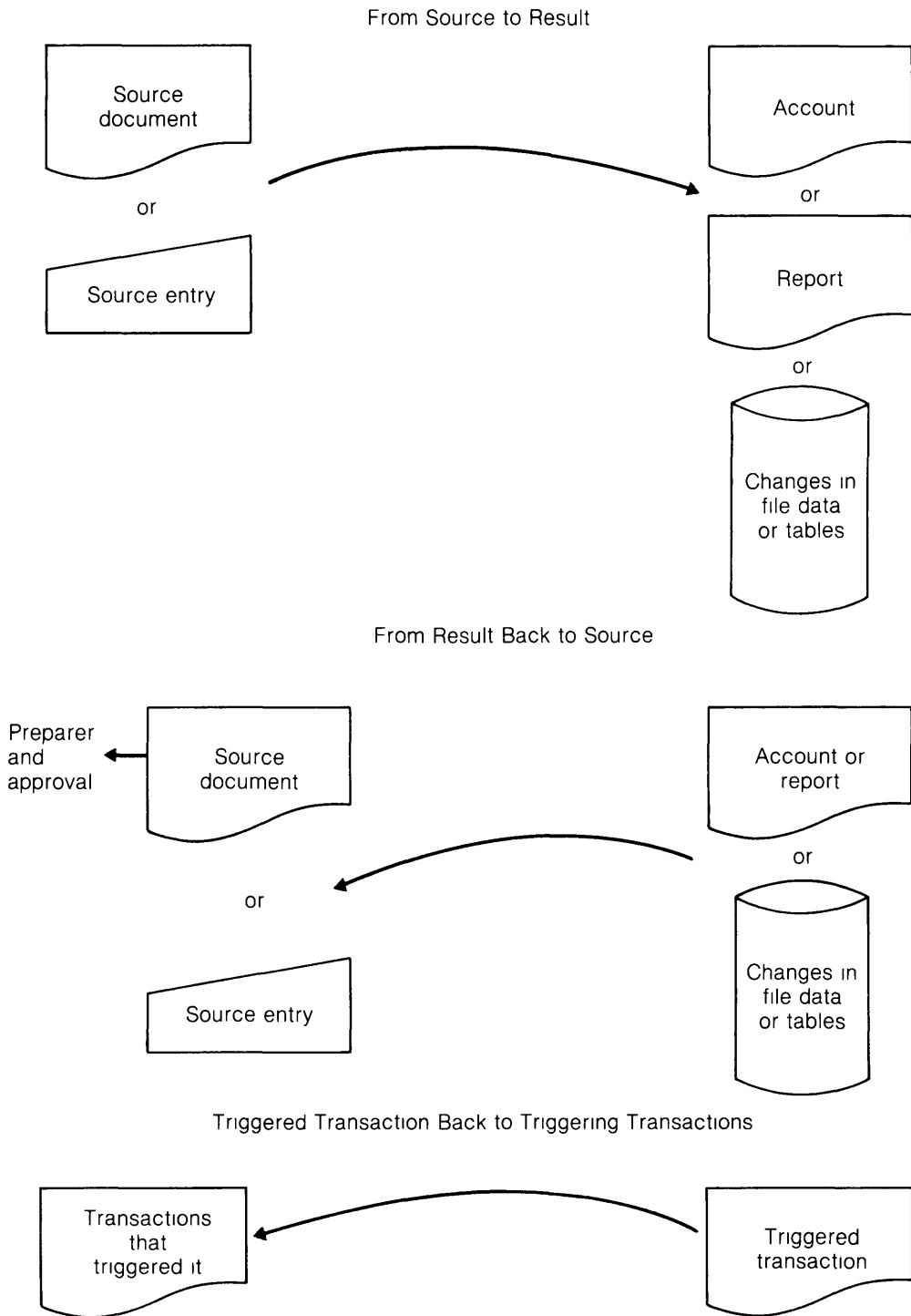
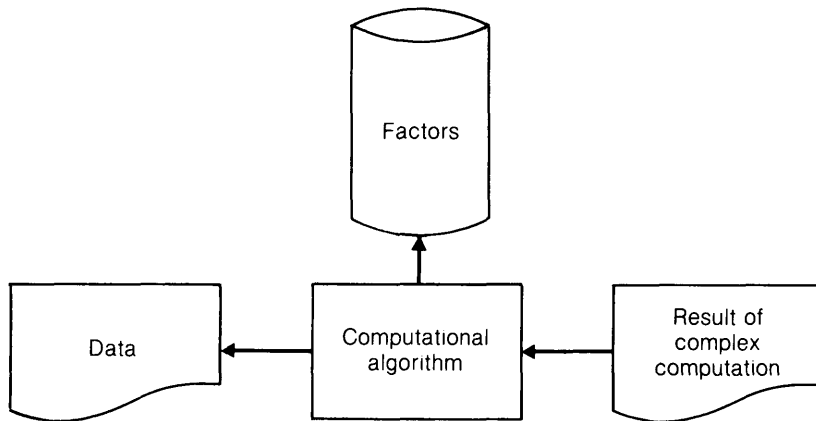


Figure 12-1 (cont)

Data necessary to recompute or review complex computation



Data necessary to identify items making up a sum



- From the ledger balance or processing total back to the set of transactions that make up the result or to the set of transactions that caused a change from a previous balance or total
- From the result back to input data items, factors, and processing logic that produced the result

The audit trail may be termed a processing or management trail since its function is to trace processing in response to internal operational questions. The following examples illustrate the necessity of the trail in day-to-day operations

- A customer requests copies of the invoices listed on the billing
- A manager requests a detailed list of all items making up an expense total on the profit center report
- A manager in purchasing requests information on how certain inventory items are used
- A customer has a record of having made a payment that has not been recorded on his account. There is a need to trace the posting of the payment to determine which account received the credit
- A manager in a foreign subsidiary questions the exchange adjustment figure

## 186 The Audit Trail in Computer Data Processing

The auditor also has a significant interest in the audit trail, since the audit trail may be used to verify that processing is carried out correctly in terms of transaction classification, processing into accounts, summarization, and so on. The auditor may also need to trace an amount back from the accounting records to the source of the transaction, as in the following procedures:

- Checking of authorization for a transaction
- Examination of the documentation for a transaction
- Examination of the transactions making up a summary account total (for example, individual transactions in a "repairs" account) to evaluate compliance with the company accounting policy

### Elements of an Audit Trail

The four elements of an audit trail in computer data processing are the transaction source record, the reference and control list, process tracing data, and transaction source references in records and accounts resulting from or affected by the transactions

#### TRANSACTION SOURCE RECORD

The three types of source records for a transaction are documents, source listings of transaction data entered directly at a terminal, and source records on magnetic storage media.

The transaction source records may be traditional documents prepared when a transaction is recorded, such as customer order documents, purchase orders, receiving documents, and payment vouchers. On the other hand, the source record may be a copy of the transaction document produced when data were entered at a terminal, examples are customer order documents and customer sales documents produced by the terminal.

Source documents must be accessible for reference. If batch processing is used, each batch should normally be assigned a batch number, and each set of documents should be filed by batch number. To locate a source document, the investigator locates the batch of documents and searches through it to find the desired item. Final identification is made by reference to name, account number, amount, and so on. Batches must be small enough to allow this search to be performed efficiently. Documents used for terminal input can be filed singly by period or by terminal, or they can be grouped and filed by period, by terminal, or by type of transaction. Source documents can also be filed by customer or employee name or number or by sequence number. In some cases, there will be a processing batch copy and a copy for the recordkeeping unit.

When transactions are entered at a terminal, there may be a pre-existing source document, or the terminal may prepare one. In some applications, no original source document is prepared, instead, a list of the transaction data entered at the terminal serves as the source document. The data items on the transaction list are identical to the data items on a traditional source document.

It is usually necessary for each transaction entered through a terminal to be assigned a unique reference number. The number, normally assigned by the computer program, identifies the transaction on transaction documents, on reference and control lists, and in documents, records, and accounts. The source listing indicates the terminal, the preparer, day, time, and so on.

Although source documents or source entry lists are usually desirable, in some cases the source entry records may be retained only on magnetic media, such as cassettes, diskettes, or magnetic tape. In such cases, a search to locate a source entry record must use a terminal or other computer facilities.

The data items in the source documents or source lists required for audit trail purposes include the type of transaction, the record or account, the amount, the processing reference, and identification of the preparer and approver of the transaction.

The person who prepared the transaction can be established by identification on a source document (initials, name, number, or stamp), work assignment records, terminal sign-on code, or terminal access records. The audit trail for an item that is entered through a terminal must include either the terminal identifier provided automatically by the hardware or the unique identifier of the preparer entered with the transaction. Both codes may be used for control and the audit trail, since it may be useful to control both the persons allowed to prepare certain transactions and the terminals on which the transactions may be entered.

The person approving a transaction is frequently included on the source document or implied by the organizational structure. The approval may often come after the fact, as transaction records are reviewed. If the transaction is entered directly at a terminal and specific approval is required, an approval code may be included in the record.

## REFERENCE AND CONTROL LIST

A reference and control list is a list of transactions processed during a processing cycle or a given period of time. It is useful for reference, search, and so on. Although a list of all transactions in the order they occur can be used for reference, such a list is usually inconvenient to search if many inquiries must be answered. Therefore, special lists are normally prepared to aid in reference or search. In accounting terms, these lists are frequently equivalent to special journals in which transactions of a given type are listed and in which subtotals and totals are computed for posting to summary or control accounts and for other control procedures. Physical batch lists and logical batch lists are the general types of reference and control lists (figure 12-2).

A physical batch list contains a listing of all transactions in a processing batch. The list contains a batch total for posting and control procedures. The batch has a number that is used as a processing reference in documents, accounts, and so on. The batch listing may contain individual source document references for the transaction, or the transactions in the batch may be identified by other characteristics, eliminating the need for separate document identifiers. Because the physical batch normally contains only one type of transaction, the batch list prepared from it is a useful source for reference and search. In order to make the list even easier to use, it may be sorted prior to output by account number order, in alphabetic order by name, or in some other sequence.

When transactions are entered as they occur at terminals, there are no conventional batches that can be used as the basis for a reference list. The transactions for a given period (such as a day, week, or month) should be stored and sorted into logical groups. The transactions should then be listed as logical batches with their own batch totals. There may be several ways to sort the transactions, so more than one set of listings may be prepared using different sort keys. Some examples of bases for logical batch listings are preparer, terminal, transaction type, account, and customer.

The data for each transaction on the reference and control list include the items normally found in special journals (source reference, account, amount, and explanatory text). Control information, such as number of items and total of amounts, is also

<b>A</b>	
Reference and Control List	
	Date XX/XX/XX
	Batch no. XXXXXX
Details of each transaction	Amount
_____	_____
_____	_____
_____	_____
_____	_____
n Records	Total \$X,XXX.XX

Reference and control list for physical batch (A)

<b>B</b>	
Reference and Control List	
	Date XX/XX/XX
---Transactions at terminal NNN	
Details _____	Amount
_____	
_____	
_____	
Total	XXXX

<b>C</b>	
Reference and Control List	
	Date XX/XX/XX
---Transactions by account type	
Type	
XXX -- -- -- --	
XXX -- -- -- --	
Subtotal	XXX.XX
XXX -- -- -- --	
XXX -- -- -- --	
Subtotal	XXX.XX

Post batch logical reference and control listing with more than one list (B,C)

**Figure 12-2** Reference and control lists

included. If the source reference is to a terminal entry, there should be some means to identify the terminal and preparer of each transaction. The date (and perhaps time) for each transaction should be included.

**PROCESS TRACING DATA**

The processing trail must allow processing to be traced. An investigator should be able to replicate the processing in order to review, audit, or reverse the processing that was performed. In the case of simple transactions, no special tracing information may be necessary. In other cases, documentation for the application may be sufficient to allow an auditor to trace the processing using the data in a reference and control list. In certain situations, however, additional tracing information may be needed. This information may be contained in special lists, it may be recorded as a special analysis attached to a journal voucher as documentation for the voucher, or it may be added to the data in the reference and control list for the transaction (see table 12-1). Some examples of cases in which process tracing information is desirable follow:

- Triggered or derived transactions
- Composition of balances when items are not otherwise listed
- Result of complex computation using factors not otherwise available
- Database, master file, and table changes

Triggered or derived transactions follow from the effects of other transactions (for example, a purchase requisition or manufacturing order triggered by an issue or sale from inventory). Tracing information for the purchase or manufacturing order might be usage data for the stockkeeping unit, inventory balance at the beginning of a review period, and receipts and issue transactions since that date. Another example of this kind of transaction is a "termination of further credit" letter triggered when a customer exceeds the agreed-upon credit limit. Tracing information for the credit termination

Need for Tracing Information	Information Source
To investigate triggered or derived transaction	Balance plus list of transactions
To verify total not subject to control total checks	List of items included in sum
To verify complex computations	Listing that includes input data, stored factors, intermediate results, and computed results
To review, audit, and/or restore changes in database, master file, or tables	Listing containing— <ul style="list-style-type: none"> <li>● Who made change</li> <li>● When change made</li> <li>● How change made (program)</li> <li>● Data item name                         <ul style="list-style-type: none"> <li>Before image</li> <li>Change</li> <li>After image</li> </ul> </li> </ul>

**Table 12-1** Needs for and sources of tracing information

## 190 The Audit Trail in Computer Data Processing

might include the customer's average balance, the balance at the beginning of a review period, and any purchases and payments made since that date. In both of these examples, the most recent transaction triggered another transaction, but tracing the events that caused the transaction requires that all recent transactions be available.

A transaction may be derived directly from another transaction, as, for example, a commission payment from a sale. Such transactions are generally handled by adding data items to the reference and control listing for the original transaction, for example, adding a commission data item to a sales list, or by creating a separate report obtained from the original data and related to the original reference and control report totals, for example, issuing a commission report or list in which the totals agree with the commission total on the sales list.

Sums of items that cannot be tested for correctness by reference to control figures or ledger balances frequently need a list of the items that are included in the sums. For example, research conducted by a government agency may require a company to supply the number of Hispanics it employs. In order to allow for a trace of this information, which can be prepared from the payroll records, the names of the employees might also be listed.

The result of complex computations generally should be supported by tracing information that would allow the result to be independently computed or at least estimated. The result of complex computations should echo all input for the computation. In addition, stored factors should be listed, and the intermediate results should be shown, if appropriate. An example for which all these supports would be necessary is a multinational company that makes complex exchange computations using exchange rates for different dates. Tracing information would include each major item to be converted, the date of conversion, the exchange rate used, and the converted result.

Database changes need to be under very strict control. It is important to be able to trace all changes in the database and to identify the person and terminal initiating the change, the time of the change, the program version used (if more than one can be used), the transaction data, and the before and after images of the fields being altered. The concept of before and after changes is also useful in making changes to relatively permanent data in master files and to tables used in transaction processing. Tracing processing for a given period is facilitated if there is a complete database, file, or table listing from which to start the trace.

### **TRANSACTION SOURCE REFERENCES**

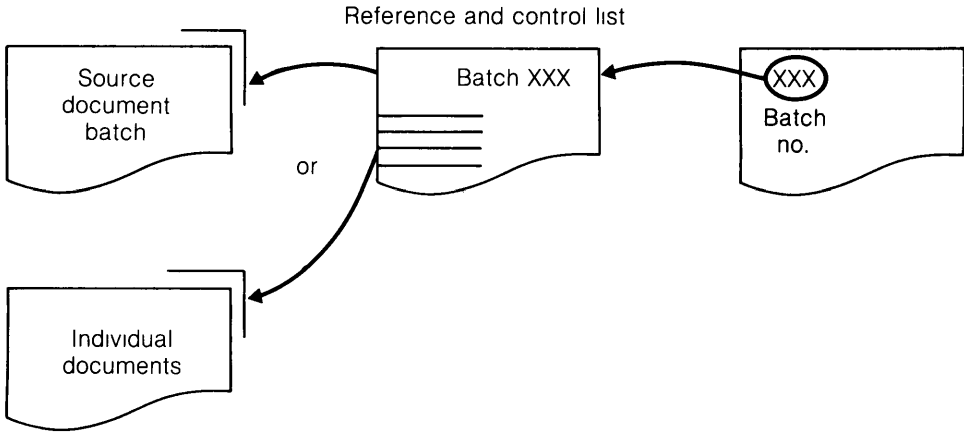
Transaction source references are used in transaction documents, records, ledger accounts, and journal entries to refer back to the source of the transaction. The transaction source reference can take several forms (figure 12-3). The most common forms are as follows:

- Reference and control list as a physical batch reference
- Reference and control list as a journal
- Transaction document identifier
- Transaction terminal entry identifier

The physical batch reference allows an investigator to go to the batch reference and control list and to the source documents, which may be stored as a batch or stored by some other reference contained in the batch listing. The batch reference may be only a sequential number, or it may be a number containing some other identifying data, such as date or type of transaction.



Reference to batch listing (reference and control list)



Reference to total obtained from reference and control list



Reference to individual document

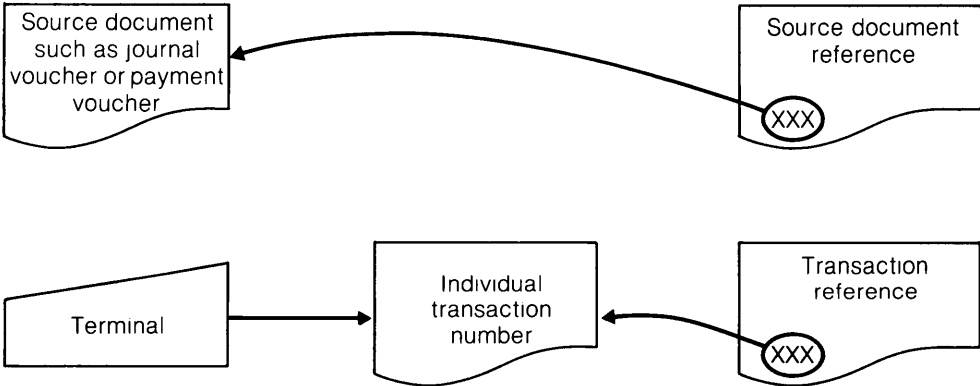


Figure 12-3 Transaction source references

## 192 The Audit Trail in Computer Data Processing

Many of the reference and control lists are special journals, therefore, the totals may be the basis for an entry to a summary account or control account. For example, the accounts receivable credit comes from the reference and control list of cash receipts (*cash receipts journal*), the posting of debit to commissions expenses comes from credit sales reference and control lists (*sales journal*), the liability credit for income tax withheld comes from payroll reference and control list (*payroll journal*). When a subsidiary ledger, such as employee pay records, customer account records, or inventory stockkeeping records, is used, the journal shows the individual accounts and the postings to be made to each. In other words, the payroll reference and control report has individual transactions, each of which represents a posting (or updating) of an employee record in the payroll master file. The references for each of these postings in the subsidiary ledger file records are to the reference and control list and through that list to the batch of transactions.

The processing reference may be to individual documents rather than to the batch. For example, a *journal voucher* may be given a *journal voucher document reference* rather than a reference to the batch in which the voucher is processed.

When transactions are entered at a terminal and an identifier is assigned by the computer, the identifying number may be used as the source reference. The identifying numbers may be in sequential order, or they may contain some coded data plus a sequence number. In either case, a knowledge of the number sequences allows an investigator to trace back to the reference and control listing and to the original entry in order to identify the preparer and the person authorizing the transaction.

A ledger account must have a record of all postings to the account. A common method for associating an account and the posted transactions is to have an account header (or master) followed by records of all transactions that have been posted. An account might have the following information:

- Account identifier (number)
- Account description
- Current account balance
- Current status of each of the following transactions
  - Debit or credit
  - Date of transaction
  - Transaction reference
  - Amount

Possible alternatives for maintaining the transaction records are as follows:

- Keep a record of every transaction since the last annual closing
- Keep a record of every transaction since the last time a listing was produced. Totals from that listing must also be stored.
- Keep only the most recent transaction and a reference to the reference and control list for that transaction. This list will contain a reference to the prior transaction, which provides a chain of reference to all transactions affecting the account. The method has serious disadvantages when an analysis of the account is required, however, when infrequent changes are required, it may be useful. For example, when pay rates change, the most recent change reference may be given and the reference and control list produced at that change may contain the reference to the prior change in rate.

## Examples of Audit Trail Methods

The methods for meeting the requirements for an audit trail are very diverse, and it is impossible to describe all variations. Three examples illustrate the ways in which audit trail requirements may be met.

Hourly payroll processing using batch processing

On-line sales order entry

Aging of accounts receivable

In each case, the applications are simplified in order to focus on the audit trail.

### HOURLY PAYROLL PROCESSING USING BATCH PROCESSING

In an hourly payroll application, the hours worked and the changes to the master file records (additions, terminations, rate changes, and so on) are the two major sources of input. The source input is on source documents that are batched for processing and then stored as a batch (figure 12-4). The batch reference incorporates the payroll date and batch number if there is more than one batch of source documents per payroll. The following output items from payroll processing contain audit trail references.

- Report of payroll file changes (containing process tracing data)
- Payroll journal (a reference and control report), which contains
  - Individual transaction data, including reference to employees (and source documents) by employee I D , name, and payroll check number
  - Derived transaction data, such as income taxes withheld (The number of dependents is part of the payroll journal, the rate table is printed at frequent intervals for review and reference )
  - Totals for posting to accounts (labor expense, FICA expense, and so on)
- Payroll checks (The audit trail reference is to the payroll journal )
- Other analyses and reports (The audit trail references depend on the data being used )

### ON-LINE SALES ORDER ENTRY

In some on-line order entry systems, operators at terminals with telephone headsets receive orders over the telephone and key the orders directly into visual display terminals connected to the computer (figure 12-5). No handwritten document is prepared. The computer program performs immediate validation of the entered data and immediately processes the order, so that the customer receives an immediate response regarding availability, shipping date, and amount of the order.

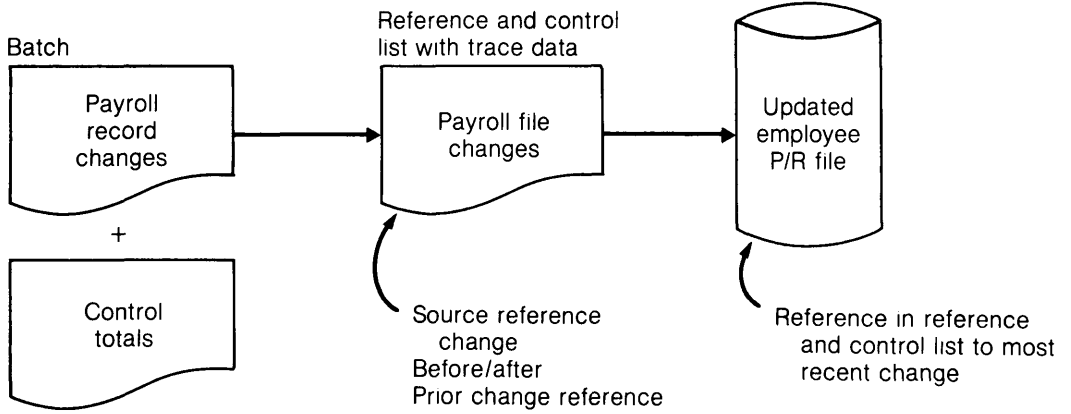
When the order is entered, the computer program assigns an order number to be used for reference purposes.

The customer is provided the reference number in case there are inquiries about the order.

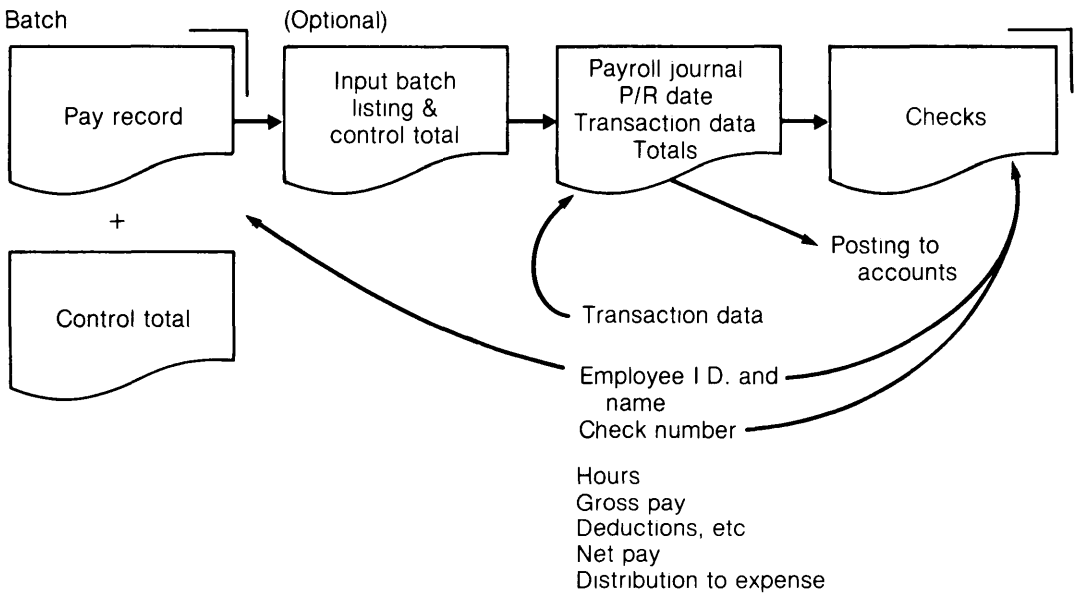
The visual display device does not produce a printed record, but a computer log is created at the time of each entry. The log can be searched and an order retrieved, with the order number or customer number serving as the search key.

A separate printer in the warehouse immediately produces a document set consisting of a customer acknowledgement, a warehouse copy, a bill of lading, a shipping

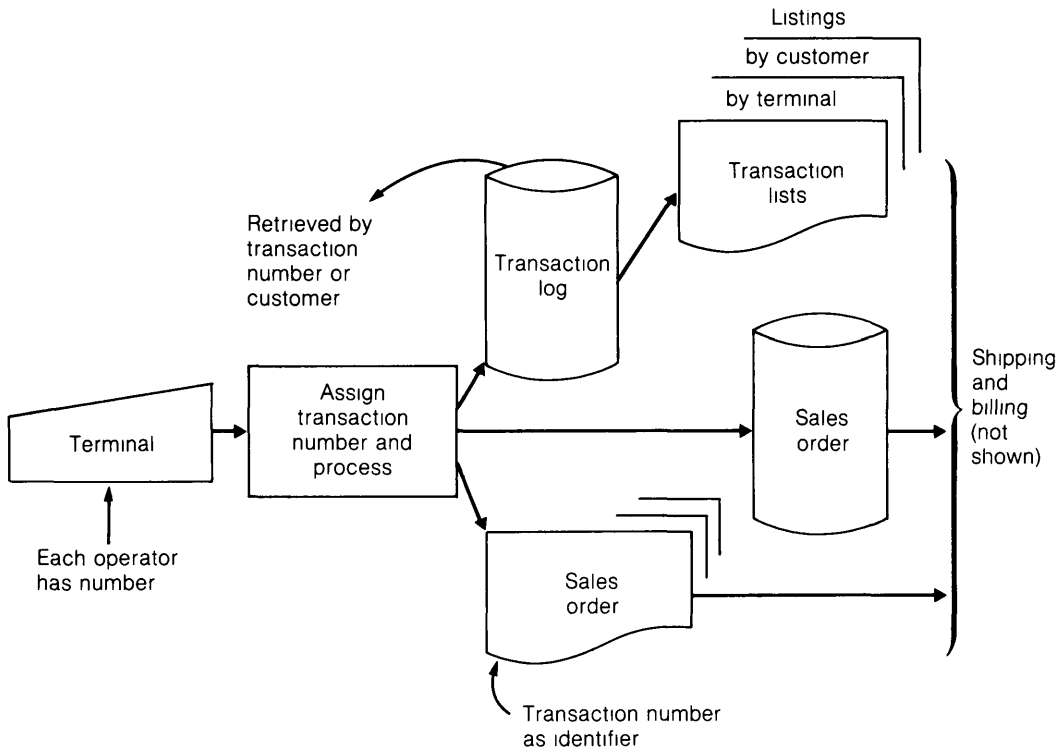
**Payroll Master File Changes**



**Payroll Transactions**



**Figure 12-4** Audit trail in batch processing of hourly payroll



**Figure 12-5** Audit trail in on-line sales order entry

copy, and a sales order (filed by customer) The document sets are used for packing and shipping the order

A file of all sales orders in process is used for billing purposes When the order is shipped, the shipping department uses a terminal to add freight charges and to make any changes, such as substitution, short shipments, and so on The computer removes the completed order from the sales order file and transfers it to a billing file Billing data items are added, and invoices are then prepared in a batch run and mailed

The sales order entry log is available on magnetic medium for reference during the day Each evening the log is processed with the following listings

- Source entry listings in the order entered (transaction sequence number)
- Reference and control list (logical batch listing by logical terminal)
- Reference and control list (logical batch in customer order)

The transaction number assigned at entry is used to identify the transaction on reference and control lists and may be used to retrieve information from the transaction entry log file The transaction number is used on the sales order document and is also used on the invoice

### **AGING OF ACCOUNTS RECEIVABLE**

The data from aging of accounts receivable may be used for the journal entry for the allowance for doubtful accounts, for management reports, or perhaps for stockholder reports The data needed for these entries or reports consist only of three or four sums for the aging categories If only the sums of the aging categories are given, an auditor

196 The Audit Trail in Computer Data Processing

wishing to trace the sums back to the individual data items would need to recompute the aging of the receivables. In some cases, the availability of the logic for recomputation is sufficient audit trail, but it can present some problems. If recomputation does not yield the same results for the aging categories, it may be difficult or impossible to reconcile the two results.

An alternative form of audit trail is to list the items (identifier and amount) in each category. This may be quite difficult if the file is large.

Another alternative is a detailed listing for only the oldest categories (figure 12-6). The listing of the older invoices is very useful for analysis and management follow-up, and it provides useful trace data to reconcile computations. The total of the aging categories should be the same as the accounts receivable control. If the file is subdivided in any way, with control figures for subgroups, then the tracing data for aging should also be subdivided so that the aging in each subdivision can be compared with the subdivision control.

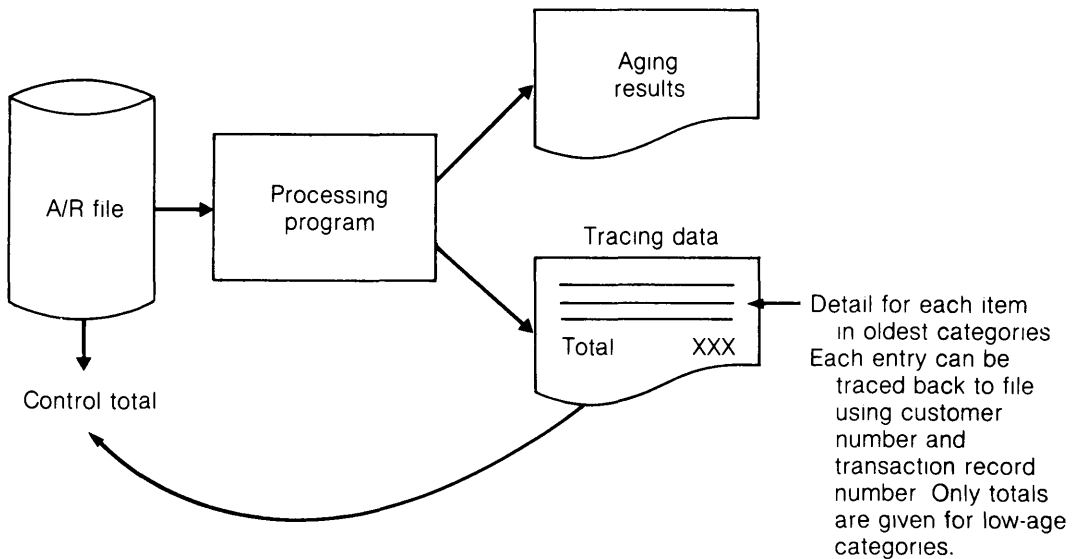


Figure 12-6 Audit trail in accounts receivable aging

## **A Review Checklist for Evaluating the Audit Trail in an Application**

A checklist incorporating the audit trail provisions appears in Appendix D of this book. The checklist may be used by the analyst during application design, by a supervisor, review group, or other analyst during the design review, or by internal or external auditor during the audit review.

The checklist is comprehensive, and not all questions are applicable to all applications. Alternatives are listed, and one or more may be included in the audit trail.

### **Summary**

An audit trail, more properly termed a processing or management trail, allows an investigator to trace processing from source data to the result or from a result to the source data in order to check processing logic and data used or to restore data items to their status prior to processing. There are four elements in a processing trail—transaction source record, reference and control list, process tracing data, and source references in documents, records, and accounts affected by the processing. There are a number of methods for obtaining these four elements.





---

## SECTION IV

# Computer Audit Tools and Techniques

Audit techniques are the procedures and methods for accomplishing audit objectives. Audit tools are the graphic, mechanical, or electronic resources used to carry out the procedures and methods. For example, sampling (an audit technique) makes use of computer programs (tools) to compute the sample size. The chapters in this section do not emphasize the distinction between audit tools and techniques because in most cases the two are so closely related that the distinction may not be important. It is assumed that the auditor has familiarity with traditional tools and techniques. Computer-assisted audit techniques use the computer and software as audit tools.

The tools and techniques for computer-assisted audit procedures can logically be divided into two types:

1. Those used after processing has been performed
2. Those used to collect audit evidence concurrently with processing

As a general rule, the after-processing techniques for obtaining evidence are more useful for the independent auditor than the concurrent-with-processing methods, which are more likely to be used by internal EDP auditors.

The tools and techniques described in the section are not equally useful nor do they receive equal use in practice. Some, such as audit software, have heavy use, others are seldom employed, however, the chapters present the full range of tools and techniques based on the philosophy that an auditor should be aware of the full set of possibilities even though he may use only a few. Also, even though some techniques find greater direct use by internal auditors, an independent auditor needs to understand their use in order to evaluate and rely upon the work of the internal auditors.

The after-processing computer-assisted audit tools and techniques are presented in three chapters. The most widely used tool, audit software, is the subject of an entire chapter, the other tools and techniques are divided into two chapters:

Chapter 13—Computer Audit Tools and Techniques Used After Processing: Audit Software Systems

Chapter 14—Computer Audit Tools and Techniques Used After Processing: Testing Files, Static Testing of Programs, and Simulation Testing

## 200 Computer Audit Tools and Techniques

Chapter 15—Computer Audit Tools and Techniques Used After Processing: Review of System Operations, Code Review, and Other Software Tools

Chapter 16 describes computer-audit techniques used concurrently with processing. In other words, these techniques collect audit evidence during processing rather than examining the system and evidence of its processing at a subsequent time.

Database management systems, the software systems for managing access to and use of databases, provide an increasingly common environment in which audit procedures must be performed. Chapter 17 deals with the use of the database software facilities to perform various audit procedures and describes special audit problems that may occur.

---

# Computer Audit Tools and Techniques Used After Processing: Audit Software Systems

# 13

Generalized audit software is the computer-assisted audit technique most frequently used by auditors today to perform common audit tasks involving EDP records. The term "generalized audit software" refers to the software packages that perform audit tasks involving records stored in computer systems.

The software reads data from a file, makes computations, and prints a report. Although the auditor may be able to perform these tasks manually or to write a computer program to do them, a generalized audit software package may accomplish the tasks more efficiently. The auditor specifies the functions to be performed by using audit software instructions. The audit software package then performs the processing steps to accomplish the tasks. Thus, by supplying only a few instructions, the auditor accomplishes tasks that would have taken a great deal of time to perform manually or to code in a programming language.

There are several advantages in using generalized audit software. It usually does not require that the auditor have a detailed understanding of hardware and software features. In general, it is easier and faster to learn than a standard programming language. Finally, the functions it provides are oriented to audit tasks, so the amount of coding and the time required to develop the auditor's program are less than that required in a general-purpose language such as COBOL or RPG.

## Functions Performed by Audit Software

The seven major categories of functions provided by generalized audit software are as follows:

- 1 Data access
- 2 Selection and statistical functions
- 3 Computation
- 4 Comparison
- 5 Record handling
- 6 Output
- 7 Miscellaneous functions

## DATA ACCESS

The most important function of generalized audit software is the capability to access the information stored on computer files (See figure 13-1 ) Data can be stored on a variety of devices and media using many different types of file organization structures and data formats Audit software packages can access the most common of these structures and formats, however, they cannot access all of them If the audit software package lacks the capability to access the storage medium, file structure, or data format of the records needed by the auditor, the data records must be converted by a manufacturer's utility program or a special interface program to a form that the audit software package can handle.

Most generalized audit software packages can access information stored on punched cards, magnetic tape, and disk Some packages can also process files stored on diskettes and certain mass storage devices, such as drums

Usually, the audit software can access sequential, index sequential, and direct access files with fixed, variable-length, or undefined records The common data formats that audit software can access are binary, alpha-numeric, zoned-decimal, and packed-decimal formats Although some audit software packages have interfaces to a few of the most common database management systems, databases cause unique problems for the use of audit software (See chapter 17 )

## SELECTION AND STATISTICAL FUNCTIONS

The selection functions allow the auditor to select data items for testing and to evaluate test results The auditor can select records in several ways

*Criteria selection* The auditor can specify criteria so that only certain records are selected for printing, which will aid in examining records for consistency and correctness For example, the auditor can specify printing of all the records of an accounts receivable file that have balances in excess of the credit limit (Figure 13-2 illustrates audit software reports based on criteria selection )

*Sampling* The types of sampling techniques provided by audit software packages include the following

- Interval or systematic sampling The auditor can instruct the audit software to select items according to a particular pattern For example, the audit software can be instructed to select every twentieth record
- Random systematic sampling The auditor can instruct the audit software to select items following a particular pattern with random starting points
- Other statistical sampling functions The auditor can also use such techniques as random sampling, stratified random sampling, estimation sampling, discovery sampling, and sampling in which the probability of selection is proportional to the item's recorded amount

Often, the audit software has facilities to help determine the sample size, to calculate means and standard deviations, and to help perform analyses of the sample results for the sampling techniques <sup>1</sup> Different sampling methods can be used together, for exam-

---

<sup>1</sup> The use of these selection techniques and the circumstances in which they can be applied are discussed in detail in Donald Roberts *Statistical Auditing* (New York: AICPA, 1978)

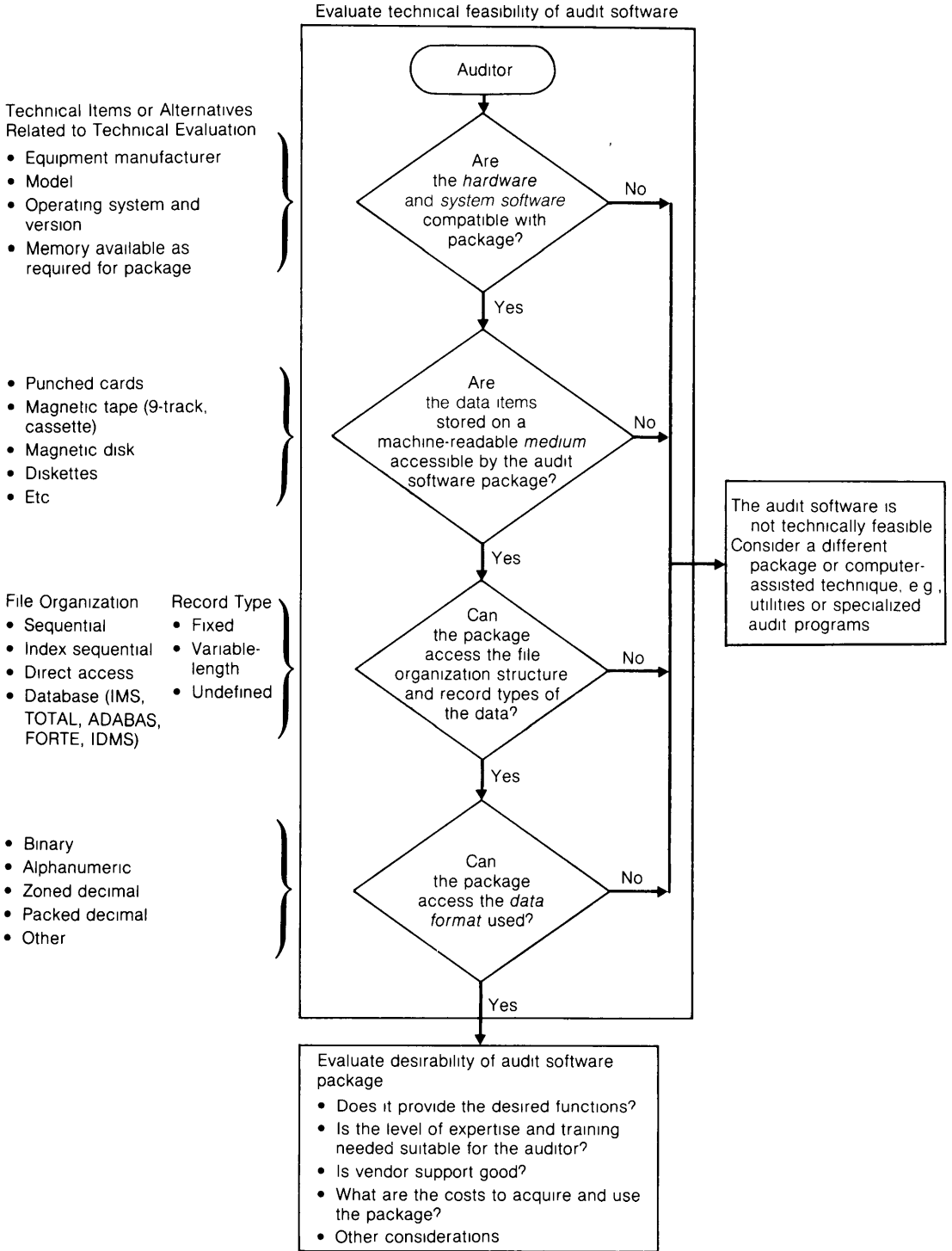


Figure 13-1 Access considerations for audit software

SAMPLE COMPANY		CPA FIRM		12-31-X2		01-15-X3		AICPA COMPUTER AUDITOR		PAGE 1	
COMPUTER COMMON AUDIT SOFTWARE - VENDOR		INVENTORY		INVENTORY		REPORT 05		RUN 1		TIME 12-46	
INVENTORY OUT OF STOCK											
PART NUMBER	DESCRIPTION	CURRENT PRICE	SALES	QUANTITY SOLD	19X2						
317	ACME REPRODUCERS	250.00		150							
331	SORTER	15,000.00		500							
COUNT	2	15,250.00									

---

SAMPLE COMPANY		CPA FIRM		12-31-X2		01-15-X3		AICPA COMPUTER AUDITOR		PAGE 1	
COMPUTER COMMON AUDIT SOFTWARE - VENDOR		INVENTORY		INVENTORY		REPORT 04		RUN 1		TIME 12-46	
DUPLICATE NUMBERS ON Y.T.D. SALES FILE											
PART NUMBER	DESCRIPTION	QUANTITY SOLD	CURRENT SALE PRICE	DATE OF LAST ACTIVITY							
302	ACHE BOND FEED WHEELS	75	2.00	1230X2							
312	AJAX REPRODUCERS	650	260.00	1211X2							
COUNT	2	725									

Figure 13-2 Audit software criteria reports

ple, selecting all records over a specified amount by criteria selection and selecting a percentage of the records less than the specified amount by interval sampling

## COMPUTATION

The computation functions allow the auditor to test the accuracy of the client's calculations and to perform quantitative analyses on the client's data. The software provides the customary arithmetic operations of addition, subtraction, multiplication, and division. For example, the auditor can use these functions to recalculate the extensions of inventory items or to recompute depreciation amounts.

## COMPARISON

Generalized audit software packages usually allow the following four types of comparisons:

- 1 Comparison of two data items within a record
- 2 Comparison of a data item and a value supplied by the auditor
- 3 Comparison of a data item and a computed value
- 4 Comparison between data items on two different files

The logic functions of EQUAL, NOT EQUAL, LESS THAN, GREATER THAN, AND, OR, and NOT are used in comparisons. Data items that meet the auditor's criteria are processed according to the instructions specified by the auditor; the other data items are not processed. For example, the auditor may test for obsolete inventory items by determining quantities on hand GREATER THAN units sold during a specified period. The potentially obsolete items would be printed, but the items with quantities not greater than units sold would be bypassed. In figure 13-3, the date of last sale of an inventory item is compared with *December 19X1* to identify items not sold in *19X2*, and the inventory turnover is calculated and compared with *3* as turnover value.

The auditor can also use the comparison function, for example, to match cash disbursements with the disbursements applied to the accounts payable file or to compare accounts receivable balances at two dates and match the differences with the sales and cash receipts files.

## RECORD HANDLING

Record handling functions are general data processing housekeeping functions that put data into a form convenient for the auditor to use. Record handling functions include resequencing the data, combining data from several files, sorting, and summarizing or consolidating the records.

## OUTPUT

Output functions create printed reports and generate computer output files. Many generalized audit software packages format the auditor's printed reports automatically unless the auditor specifies column widths, editing, spacing, and so on.

In addition to the capability to print reports with tabulations and listings, many packages have a *forms* or *confirmations* function that enables them to print confirmations or form letters and combine them with data from other files.

Output files in machine-readable form (for example, tape or disk) are useful when the auditor plans to perform additional audit procedures on data already selected or computed. Many packages can generate several reports or output files from a single processing of the client's file.

PART NUMBER	TY PE	QUANTITY SOLD 19X6	QUANTITY AT 12-X6	EXTENDED AT X2 STD	URNS IN X2	DATE OF LAST SALE	AUDIT COMMENTS
303			200.00	12150.00	0.00		NOT SOLD IN 19X2
304			300.00	18150.00	0.00		NOT SOLD IN 19X2
316			135.00	135000.00	0.00		NOT SOLD IN 19X2
326			75.00	187000.00	0.00		NOT SOLD IN 19X2
330			100.00	88100.00	0.00		NOT SOLD IN 19X2
TYPE COUNT		5		440400.00			
313	02	600.00	40.00	15000.00	12.90	0630X2	LAST SOLD PRIOR 9-X2
TYPE COUNT		1		15000.00			
301	03	7000.00	0.00	0.00	2.00	0830X2	TURNED LESS THAN 3
302	03	825.00	1000.00	2225.00	.50	1231X2	TURNED LESS THAN 3
305	03	2500.00	400.00	40400.00	7.14	0101X2	LAST SOLD PRIOR 9-X2
307	03	25.00	10.00	902.50	2.00	1115X2	TURNED LESS THAN 3
310	03	200.00	25.00	7500.00	4.00	0831X2	LAST SOLD PRIOR 9-X2
328	03	80.00	10.00	20000.00	1.60	1030X2	TURNED LESS THAN 3
TYPE COUNT		6		71027.50			
COUNT		12		526427.50			

Figure 13-3 Audit software report using computation and comparison functions



## MISCELLANEOUS FUNCTIONS

Other functions often performed by audit software include the following

- Table lookup, which aids in categorizing data or translating unusual code structures
- Job control language generation (This software function automatically creates the job control instructions for the computer's operating system. These instructions are required for the audit program to be run. This function reduces the need for the auditor to know the job control language.)
- Bar graph analysis, which creates reports in the form of histograms (figure 13-4)

## Types of Generalized Audit Software

Generalized audit software systems can be grouped according to (a) how the auditor expresses instructions to the audit software package (coding versus checklist) or (b) how the package translates the auditor's instructions to the computer (precompilers versus interpreters).

### CODING VERSUS CHECKLIST

Auditors communicate their instructions to the audit software package in either of two basic ways: coding or checklist. Coding may be either free-form coding or use of coding specification sheets. In this method of communication, the auditor writes instructions using a set of instructions and coding rules. Coding in an audit software language is generally easier to learn and to use proficiently than coding in a programming language such as COBOL because the terminology and functions are oriented toward audit tasks. Coding is more flexible than the checklist approach, but it requires the auditor to define precisely the requirements of the audit procedure. Figure 13-5 illustrates audit software coding.

Some auditors prefer free-form coding because they like the freedom to code statements without using a specification sheet. Free-form coding also facilitates input of auditor's instructions through remote terminals. Use of specification sheets for coding, on the other hand, helps the auditor remember the format and sequence of instructions.

The auditor can also use a checklist with predefined routines. In the checklist approach, the auditor checks the functions to be performed from a set of available functions. This approach, shown in figure 13-6, can be used by an auditor who has relatively little training in audit software. The disadvantage of this approach is that the auditor is limited to those functions on the checklist.

### PRECOMPILERS VERSUS INTERPRETERS

The other categorization of audit software is based on the two basic methods by which an audit software package can translate the auditor's instructions to the computer. In the precompiler approach, the audit software package translates the auditor's instructions into programming language source code statements in a language such as COBOL or FORTRAN. These statements are then compiled and processed like any other application program.

One advantage of a precompiler package is that it enables an auditor who knows the programming language to modify the generated source code to perform functions not provided by the audit software package. The auditor simply inserts COBOL, FORTRAN, or RPG statements into the program generated by the precompiler.





<b>Comparison of Two Files</b>		
Do you want to compare files? Enter Y or N .....	<input type="checkbox"/>	32
Do you want to match the records or combine them?		
Match = M .....	<input type="checkbox"/>	33
Combine = C .....		
Do you want a message printed if the primary file has no match on the secondary file? Enter Y or N .....	<input type="checkbox"/>	34
Do you want a message printed if the secondary file has no match on the primary file? Enter Y or N .....	<input type="checkbox"/>	35
<b>Interval Selection</b>		
Do you want to select every nth item? Enter Y or N .....	<input type="checkbox"/>	36
What is the starting point? .....	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	37 44
What is the interval (n)? .....	<input type="text"/>	45
What output medium do you want? Report = 1 File = 2 Confirmation = 3 .....	<input type="checkbox"/>	46
<b>Random Selection</b>		
Do you want to select records randomly? Enter Y or N .....	<input type="checkbox"/>	47
What is the random starting number? .....	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	48 55
What is the population size? .....	<input type="text"/>	56
What confidence level do you want (percent)? .....	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	57 63
What sample size do you want? .....	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	64 71
What output medium do you want? Report = 1 File = 2 Confirmation = 3 .....	<input type="checkbox"/>	72

**Figure 13-6** Audit software using a checklist approach

Another advantage of the precompiler package that generates COBOL source code is that the program can be processed (with some modifications) on computers designed by many different manufacturers. COBOL compilers are not completely standardized, but conversion of the programs from one manufacturer's computer to another is relatively easy.

The other approach uses interpreters to translate the auditor's instructions directly into machine language. The advantage of interpreters is that programs generated directly in machine language can be processed without going through the compilation step. However, in this approach the auditor cannot add source code statements to the generated code to perform a function not provided by the software package, although most of the packages provide points in the processing of the audit software program, called "exits," where the auditor can add source code subroutines. (See figure 13-7.)

## Selecting a Generalized Audit Software Package

Audit software has been developed by the large public accounting firms and by various software vendors. In selecting a generalized audit software package, the auditor should consider several issues. Appendix E contains a checklist of selection factors.

### DEFINITION OF NEEDS

Acquiring a package can be expensive, not only in the cost of the software itself, but also in the cost of training personnel to use it. The auditor should consider the number and type of potential audit software uses. Making a list of proposed applications and indicating the functions to be performed will help to identify the extent of the need for an audit software package. The auditor can then analyze the list, considering the following questions:

- Does the application to be audited represent a material dollar value relative to the total assets or income? Is there any other reason why it is of audit concern?
- Would manual procedures meet the audit objective as effectively as the audit software? If so, would the manual procedures be as efficient?
- Does the use of audit software allow different audit procedures that would not be practical otherwise?
- Would other computer-assisted audit techniques satisfy the audit objective as effectively? If so, would the alternative techniques be as efficient?

### IDENTIFICATION OF COMPUTER RESOURCES

Many audit software packages can run on only one type of computer unless they are modified. Therefore, the auditor must choose a package capable of running on the hardware that will be used. If the auditor will use a variety of computers, an audit software package using a COBOL precompiler approach may be appropriate. Although the audit software package may have to be modified to operate on different computers, the instructions employed by the auditor generally remain the same.

Similarly, the auditor should identify the operating system, peripheral devices, and data access methods used at the installations where the audit software will be employed and compare these with the capabilities of the audit software packages being considered.

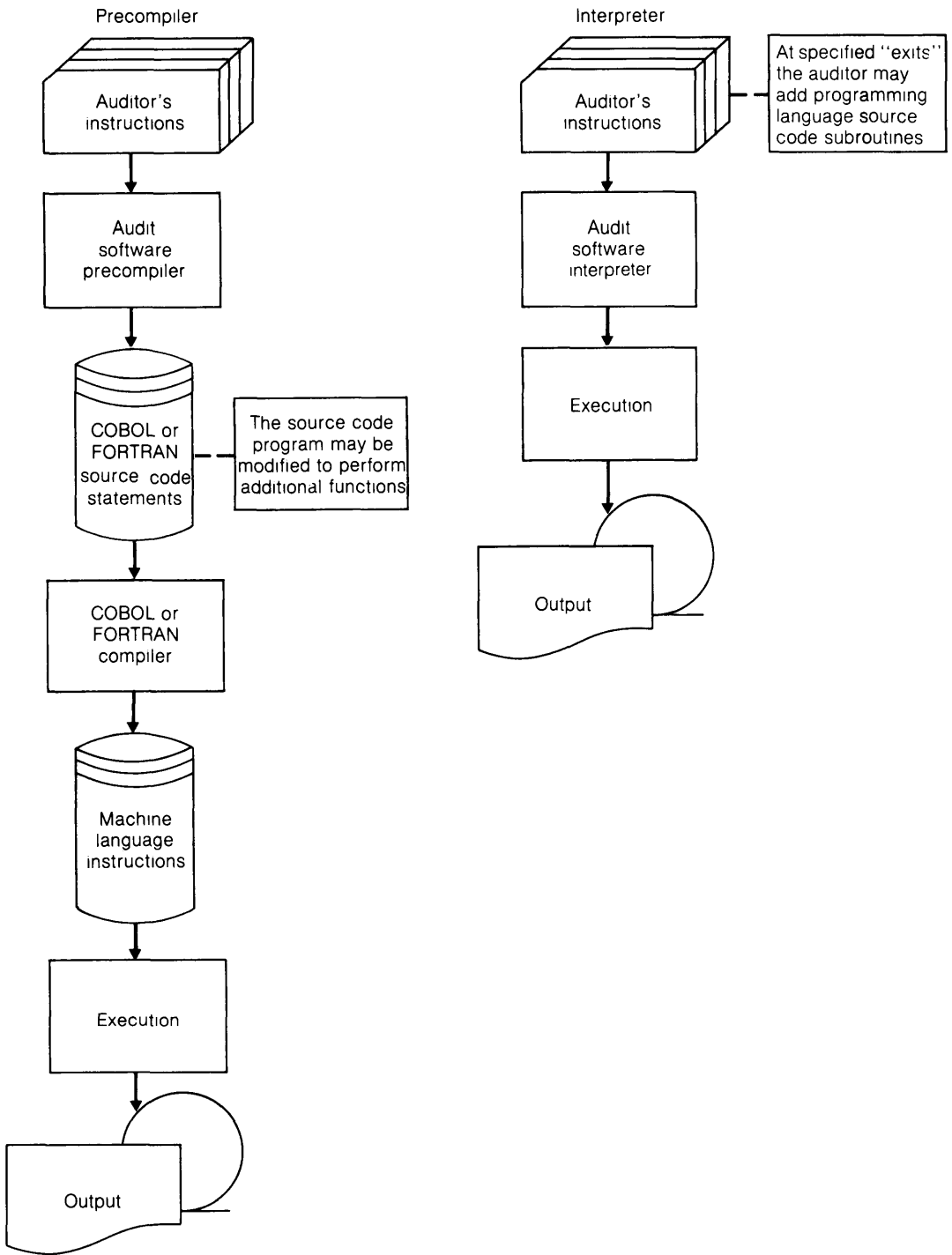


Figure 13-7 Precompilers versus interpreters

The auditor should also consider the data processing environment. Although many auditors run their audit software at the installation where the records under audit are kept, some auditors process the client's records at a service center, on a time-sharing system, or at the CPA's own installation. These environments help solve the problem of acquiring audit software that runs on different systems because all the audit applications are run at one installation. If necessary, the files can be converted to the format required by the audit software installation.

## **LEVEL OF AUDITOR EXPERTISE REQUIRED**

Different audit software packages require varying levels of expertise. The expertise involves knowledge of computer concepts and data processing in general and training in the use of the audit software package itself.

### **General EDP Knowledge**

Some packages require that the user possess a thorough background in data processing. These packages are often very flexible and allow a wide range of applications. They are often designed to be used by computer audit specialists or by data processing experts working with an auditor. On the other hand, some software packages are less complicated to use but offer fewer capabilities and options. In some instances, the auditor only checks a box next to the function to be performed. Most of the available packages fall somewhere between these extremes. For these packages, the auditor need not have an extensive technical background, but he should have a good understanding of data processing concepts and techniques.<sup>2</sup>

### **Training in the Use of the Audit Software Package**

The extent and depth of training in the instructions and use of a particular audit software package depend on the following three factors:

- The user's level of general EDP expertise
- The design and complexity of the package
- The availability of assistance

In selecting a package, the auditor should consider these factors in determining whether the available training is sufficient. If technical help is easily available when the user has a problem, the training course can be shorter than it would be if the auditor using the software must be able to solve most of the problems.

The availability of technical assistance is very important, an expert in both data processing and the use of the package should be available to solve problems when they arise. There are various methods by which the auditor can obtain technical expertise. The auditor may have (or acquire) the expertise, it may be provided by the software vendor's consulting service, or the auditor may be able to consult with other users of the audit software package.

---

<sup>2</sup> Elise G. Jancura, "Technical Proficiency for Auditing Computer-Processed Accounting Records," *Journal of Accountancy* (October 1975).

**FUNCTIONS REQUIRED**

The auditor should anticipate the audit procedures to be performed with the audit software and determine whether the packages being considered provide the audit software functions needed to perform them. For example, a number of packages have minimal statistical sampling capabilities and lack a "confirmations" facility.

**VENDOR SUPPORT**

The kind of support provided by the software vendor is not a consideration that is unique to the selection of audit software, but it deserves mention here because it can be extremely important. The auditor should determine answers to such questions as the following:

- Where is technical support obtained?
- How quickly will the technical support be available?
- How much does technical support cost?
- How much effort will the vendor expend to keep the software from becoming obsolete?
- Is the vendor a reliable, financially stable organization?

**Case Example of the Use of Audit Software**

The major application of audit software systems is in audit areas where there are large numbers of computer-stored records to which audit procedures are to be applied. Some examples are accounts receivable, inventory, accounts payable, and payroll. To illustrate the use of audit software in the audit, this section presents a simplified description of the use of software packages in connection with examination of accounts receivable.

The client has an accounts receivable file for 5,000 customers. The client uses open item billing procedures, and its record-keeping for each customer consists of relatively fixed information about the customer, such as name, address, credit rating, and credit limit (amount, date of review), and variable data (such as purchases year to date, the number of unpaid invoices, and total amount unpaid). There is a separate trailer record for each unpaid invoice.

The auditor, as part of the preliminary phase of the review of the computer data processing system, gained an overall understanding of the accounts receivable application and the general EDP controls for data processing. The auditor decided to complete the review of accounting controls in EDP and in accounts receivable, since he planned to rely on the internal accounting controls. He obtained information on the compatibility of the client's computer with the available audit software package, and he established that he would have permission to use the client computer and access the files needed to perform audit tests.

The audit plan, which incorporated the results of the preliminary phase of the review, included some of the following tests:



Compliance Test

Test a customer account record for the existence of credit rating, credit limit, evidence of a review of the credit limit within two years, and balances in excess of the credit limit

Test customer account records for unpaid invoices more than three months old with balances less than five dollars

Test the write-off of invoice balances less than five dollars

Analyze credit balances by size and trace a sample of credit invoices to transaction documents

Substantive Tests

Foot the accounts receivable file by section and trace totals to accounts receivable control accounts and to general ledger total

Analyze the distribution of account balances by size and compare them with the distribution for the prior year

Age the balances due (by invoice)

Prepare confirmation requests using stratified random sampling

In using an audit software package for these tests, the following procedures are performed

- 1 The field auditor gains familiarity with the flow of processing and the generalized logic of accounts receivable processing, as well as the specific logic for handling unidentified payments, handling small differences between amount due and payment received, writing off small differences, giving credits and allowances, establishing and updating credit rating and limit, and authorizing exceptions to credit limit

Audit Objective

To test compliance with management's policy that all customer accounts should have a credit rating and a credit limit, that the limit should be reviewed at least every two years, and that the limit should not be exceeded without authorization

To test compliance with management's policy that all unpaid invoices with balances less than five dollars should be reviewed for write-off because they are generally due to minor differences between the customer and the business relative to freight, discounts, and so on

To test compliance with internal accounting controls for write-off of small invoice balances

To test compliance with management's policy and internal accounting controls for credits and allowances

Audit Objective

To check the existence and agreement of detail records with ledger totals

To identify any significant change in distribution of account balances that might reflect changes in or lack of adherence to management's policy and accounting controls

To evaluate collectibility of accounts receivable balances

To obtain customer verification of amounts due

## 216 Tools and Techniques Used After Processing: Audit Software Systems

- 2 The auditor obtains the layout of the accounts receivable file, descriptions of all codes used, and procedures for backup and recovery
- 3 He obtains a copy of the accounts receivable file at the date desired for the audit, such as period end. A copy is prepared under auditor supervision and maintained under audit control
- 4 Instructions are prepared for the audit software package. The instructions, in the format required by the package, are key-entered and verified.
- 5 The audit software package is loaded onto the client's or the service center's computer by regular operators under the auditor's supervision
- 6 Instructions are read and validated by the audit software system. Error messages are printed if the instructions are incorrect or incomplete. If there are errors, the job is aborted, and the errors are corrected and revalidated in a second run. The process is repeated until the instructions are accepted as error-free by the validation procedure
- 7 The audit copy of the accounts receivable file is mounted, and the job is executed. The printed output is taken by the auditor along with the copy of the accounts receivable file, and the outputs are incorporated in the working papers

The instructions written by the auditor include the following

### Instruction

Specify fields in the customer's record that are financial amounts and the basis for subtotaling of fields

Specify analysis of balances by size  
Specify size intervals

Specify use of aging routine for invoices. Specify aging intervals

Specify use of confirmation routine  
Specify stratified random sampling  
Provide sampling specifications

Test each record for existence of credit rating, credit limit, date of last review less than two years old, and balance less than credit limit

Test each unpaid invoice to determine whether it is more than three months old or has a balance of more than five dollars

Obtain a sample of invoices less than three months old with balances less than five dollars

Obtain a stratified random sample of credit invoice amounts greater than five dollars

### Purpose/Result

Fields specified as amounts are footed, and subtotals are prepared

Balances are analyzed by size, both positive and negative

An aging report is prepared

Confirmation requests plus a control sheet for audit workpapers are prepared

A list of exceptions is generated, with the data necessary for investigation

A list of exceptions is generated, with the data necessary for investigation

A list containing a sample of items is generated, with the necessary detail to trace to subsequent client write-off documentation

A list of sample credit invoices is produced, with the detail necessary to trace to source documents

All the output from the audit software should contain sufficient data to tie to control figures, such as number of records, ledger balance, and so on. It should have sufficient identification, such as headings and labels, for use in audit workpapers. Some of the identification may be provided by the audit software package, other identification may have to be added manually.

## Summary

Generalized audit software is the most commonly used computer-assisted audit tool. It allows the auditor to assess and work with the financial records on computer files. Most audit software packages can access data, select data from the files, make calculations, compare two files, sort and summarize files, print reports, and write new files.

Packages can be categorized by their types of audit instructions or by the method used to translate those instructions to the computer. Audit instructions can be expressed by coding or by marking a checklist. Translation of the instructions to the computer can be done by precompilers in conjunction with a source code compiler or by interpreters.

In selecting an audit software package, the auditor should consider the needs and potential uses of the software, the computer sources available, the data processing expertise available, the audit functions to be used, and the quality of the vendor's support.



---

# Computer Audit Tools and Techniques Used After Processing: Testing Files, Static Testing of Programs, and Simulation Testing

# 14

In addition to generalized audit software, auditors have developed a number of other tools and techniques to assist them in the review and evaluation of the results produced by data processing systems. Many of these techniques were not created specifically for audit purposes but were adapted from developments in systems and programming. Since the problems of testing, review, and evaluation faced by systems designers and implementors are similar to those faced by auditors, a number of the software aids that facilitate systems work have been useful in auditing. Therefore, if the audit involves a data processing system, the auditor should be familiar with available software aids.

Computer audit tools and techniques for audit testing and audit data collection after processing can be grouped into five broad categories.

1. Testing of files. This category consists of techniques the auditor might use in accessing and processing files maintained on a computer. Usually, this is performed as part of substantive testing and is directed at testing the completeness and correctness of the individual records within a file. For the most part, these techniques do not give the auditor assurance with regard to the operation of the system during the period under review.
2. Static testing of programs. This category includes the software packages and techniques the auditor might use in processing test data and observing the results. Static testing does not affect the live data being processed. The tests are conducted after processing rather than concurrent with processing.
3. Review of system operation. This category includes tools and techniques the auditor can use to obtain evidence on what has gone on in the system. In effect, these aids provide an additional audit trail. The information on system operations is generally collected for purposes other than auditing. The data records are written onto a log during system operation, and the records are available, on an after-the-fact basis, for audit use.
4. Review of program code. This technique has had limited use over the years, but in some cases, it has proven to be the only effective way to cope with certain audit problems. There are some software aids that can reduce the effort involved in program code review. These include flowcharting software, cross-reference sys-

## 220 Testing Files, Static Testing of Programs, and Simulation Testing

tems, and record layout software. Other software packages allow the auditor, after reviewing a program, to monitor any subsequent changes to the code.

5. Other software tools. This category covers other software packages that might, in certain circumstances, serve an audit purpose. Most of the software in this category has had only limited use because auditors do not know of its existence. Some readily available reference sources can be used to supply information about packages that might be useful to an auditor.

The first two categories, testing of files and static testing of programs, are covered in this chapter, and the other three categories are presented in chapter 15. Audit tools and techniques applied concurrently with processing are described in chapter 16.

## Testing of Files

A major EDP audit activity involves the performance of various substantive tests. Typical applications include the preparation of confirmations, the comparison of inventory test counts with book balances, and the aging of accounts receivable. Although generalized audit software is the most widely used tool for performing these procedures, a number of other useful approaches can assist in the manipulation and testing of computerized data files.

### QUERY FACILITIES OF FILE MANAGEMENT AND DATABASE MANAGEMENT SYSTEMS

A significant trend in data processing is a relatively easy-to-use query facility for accessing records in the computer files of the organization. The query facilities are often intended to be used by personnel who are not programmers, and the query language tends to be user-friendly, with forms and screen formats that aid in formulating requests.

The auditor should consider the use of these query facilities as an alternative to audit software for accessing client files and performing audit tests both when audit software is not available or cannot be used and when the query facilities are more efficient or effective.

### CUSTOMIZED AUDIT SOFTWARE

In this approach, the auditor writes computer programs to perform audit functions. These programs are written in one of the programming languages available on the particular hardware involved and are designed specifically to work with the files to be audited. While this approach can be quite effective, it is expensive and frequently difficult to implement.

A major problem is the difficulty of obtaining, training, and retaining personnel with the required expertise in both auditing and programming. Working with different clients increases training needed. Although languages such as COBOL are standardized, each implementation on particular hardware has a few language characteristics that are unique plus major differences in job control instructions and content of error messages. Also, unless an auditor has the opportunity to use his programming skills on a fairly regular basis, it will be difficult for him to retain a reasonable level of proficiency. Some firms use auditors trained as computer audit specialists for preparing custom audit programs, others use EDP personnel who are part of the audit teams.

Another problem encountered in using customized audit software concerns program maintenance. Systems seldom stay the same over a period of time, as a result, a computer audit program written for an audit in one year will have to be reviewed, revised, and tested before it can be used for the next year. The changing and updating sometimes make it difficult for the auditor to justify the development costs of customized audit software.

Some of the problems associated with writing customized computer audit programs may be offset by the fact that much of the processing the auditor requires is quite simple, so that the programs are not complex. For example, in one company a very time-consuming manual accounts receivable confirmation procedure was converted to computer processing by writing a special program. This approach was adopted because the record formats used on the client's file could not be manipulated by any of the generalized audit software systems. The writing and debugging of the relatively simple computer program took slightly less than eight hours.

Another development that increases the potential use of custom-written audit software is availability of very high level languages for writing programs (sometimes called fourth generation languages). The development process is much easier with these languages than with procedural languages such as COBOL and requires only a fraction of the time. Also, because the languages use very high level instructions, the auditor who writes (or supervises the writing of) a program has much more control over the development and is less likely to make errors in logic or processing procedures.

## **MODIFICATION OF EXISTING CLIENT PROGRAMS**

In order to reduce the time and effort required to create customized computer audit programs, the auditor should consider the possibility of modifying existing client programs. In many cases, the processing that the auditor requires may be similar to one or more of the application programs used within an installation. For example, if the auditor needs to print account receivable confirmations, he may find that most of the logic for such a program can be taken from the program used to perform month-end billings. The auditor can obtain a copy of the program and modify the copy to meet specific audit requirements. This approach often simplifies the task of obtaining information about file layout and organization.

## **UTILITY SOFTWARE**

Computer systems are typically supported by a variety of utility software packages provided by the hardware vendor. In many cases, the utilities will be part of the operating system. These programs are designed to perform simple, repetitive tasks. For example, there are utilities to sort and merge files and to transfer files from one format or medium to another, for example, card to tape, tape to disk, and disk to print. Other utilities can be used to maintain password tables or to perform modifications of the operating system software. Often, utility programs are used by operations personnel to correct minor operating problems. Sometimes utilities will be used to perform fairly straightforward, one-time-only jobs. Others, such as the trace utility, are useful in reviewing the logic of a program.

Utility systems offer a large range of capabilities, and most of them are relatively easy to use. Rather than teach auditors the details of the programming languages

## 222 Testing Files, Static Testing of Programs, and Simulation Testing

implemented on various systems, some audit groups have trained their personnel to use the utility systems. Auditors who have tried this approach have often found that many of the audit tasks they wanted to perform could be handled easily by the use of utility packages. For example, a disk-to-print utility may be used to prepare an unformatted listing of the contents of a disk file.

In dealing with almost any system, an auditor would find it worthwhile to have some knowledge of the available utilities. The auditor could start by obtaining from the hardware vendor a copy of the utility manual, which identifies the available capabilities. Most utilities can be used by preparing relatively simple parameter cards that provide information about file structure and processing options.

### **AUDIT SOFTWARE APPLIED TO SMALL COMPUTER SYSTEM FILE TESTING**

Generalized audit software is not available for most minicomputers and small business computer systems. Custom programs, modification of client programs, and utility software can be used to test the files in these systems, but it is also possible to take the files from the small business computer to a larger computer for audit software testing. For example, a service center can convert small computer system files into a format for use with audit software. The functions performed on the converted data files are the same as those performed by generalized audit software. Some examples of these functions are selecting accounts for confirmation, preparing confirmation requests, comparing inventory counts, calculating inventory item values, and calculating total inventory value.

In using these packages for small system file tests, the auditor copies relevant files on a diskette or other medium and forwards the data to the service center. The auditor may transmit the data through a terminal, deliver the diskette to a local office of a service center, or ship the diskette to the service's central processing facility. Specialized hardware has been specifically developed to read the major diskette formats and to translate them into a common format that can be loaded on a disk file at the service center.

Once the diskette has been loaded on a disk file, the client's records are available to the auditor. The auditor may transmit definitions of record formats and other instructions to the service center by the use of a terminal. The audit software then performs its audit tests, and the results are returned to the auditor.

Figure 14-1 illustrates a terminal session in which the auditor has asked that a random sample of invoices, having specified confidence level, precision limit, and occurrence rate, be taken from a client's file. Accounts with credit or zero balances are to be excluded from the sample. When the processing has been completed and the output has been returned to the auditor (figure 14-2), the auditor can remove the files from the system.

### **SPECIALIZED INDUSTRY SOFTWARE**

Specialized audit software has also been developed for particular industries. For example, a number of computer audit software packages have been designed to perform audit tests unique to banking. Some of these packages offer a wide range of standard audit steps that can be applied to a bank's data files. An auditor who works with financial institutions should investigate these industry software systems.



```

A/R > SEL RAN
SAMPLE SIZE: 0
STARTING SEED: 0
CONFIDENCE LEVEL (e.g. .95): .95
PRECISION LIMIT (e.g. .3): .3
OCCURRENCE RATE (e.g. .1): .1
EXCLUDE CREDIT BALANCE ACCOUNTS? YES
EXCLUDE ZERO BALANCE ACCOUNTS? YES

A/R RANDOM SELECTION

DATABASE CONTAINS 107 INVOICES
ZERO/NEGATIVE AMOUNT FILTERING LEAVES 107 INVOICES

CALCULATED SAMPLE SIZE IS 18
STARTING SEED = .154161154309898
FINAL SEED = .059780492447317

```

**Figure 14-1** Specialized audit software—Instructions to sample an accounts receivable file

```
A/R > PRINT FROM CONFIRMATIONS
```

CONFIRMNUM	ACCOUNT NUMBER	INVOICE NUMBER	AMOUNT	INVOICE DATE
1	11200	2928	\$ 0.37	SEP 26, 198X
2	11200	29276	916.75	SEP 28, 198X
3	11300	29269	143.18	SEP 28, 198X
4	12100	26951	338.83	DEC 03, 198X
5	31200	29259	757.62	SEP 28, 198X
6	31200	29260	57.57	SEP 26, 198X
7	31500	29255	500.00	SEP 26, 198X
8	32100	95996	292.86	SEP 26, 198X
9	32400	26923	41.25	SEP 26, 198X
10	33500	26920	342.57	SEP 28, 198X
11	35100	25187	12.03	SEP 28, 198X
12	52300	81516	400.15	SEP 28, 198X
13	53100	81526	440.05	SEP 28, 198X
14	54400	49200	398.53	SEP 28, 198X
15	54500	49500	759.59	SEP 28, 198X
16	54900	81415	634.35	SEP 28, 198X
17	56200	11100	653.42	SEP 28, 198X
18	57900	14121	263.00	SEP 28, 198X

```

18 RECORDS READ FROM CONFIRMATION
18 RECORDS USED.

```

**Figure 14-2** Specialized audit software—Output from accounts receivable sample

## 224 Testing Files, Static Testing of Programs, and Simulation Testing

Computer audit software packages have also been written by auditors for use in the stock brokerage industry because of the unique aspects of this industry. The auditor is required to produce some reports that are based on record formats and sequences that are not normally maintained by the client. Further, some of the output from the audit is specified by regulatory authorities. As a result, specialized audit software is especially suitable to the requirements of the stock brokerage industry.

### Static Testing of Programs

In performing a static system test on application programs, the auditor processes test data through the application programs and observes the results of the processing. This approach is used both to confirm the auditor's understanding of an application system's logic and controls and to test software to be used in performing audit tasks.

A static test is performed as a separate processing activity that is completely distinct from normal processing. The testing environment is controlled and is not subject to the schedule pressures and random events of normal processing. This has the advantage of isolating testing from operational factors, so that the testing may be concentrated on the application logic. The disadvantage is the lack of testing under operational conditions.

Static testing of application programs requires both transaction data and test files, which may be prepared manually or with the aid of test data generators.

#### TEST DATA

Test data input records consist of a variety of transactions that have been created to use all elements of application program logic represented in its documentation. Therefore, the test data records include errors and boundary values (that is, breakpoints) that will trigger alternative decisions within the program. For example, if a disbursement system is not supposed to write a check in excess of \$500, test data would be designed to include, among other things, the following:

<u>Test #</u>	<u>Disbursement Amount</u>	<u>Test</u>
1	\$499.99	Below boundary value
2	\$500.00	Boundary value
3	\$500.01	Over breakpoint (error)

Since errors frequently occur at boundary values, testing the boundary value, boundary value plus one, and boundary value minus one is effective in testing processing logic.

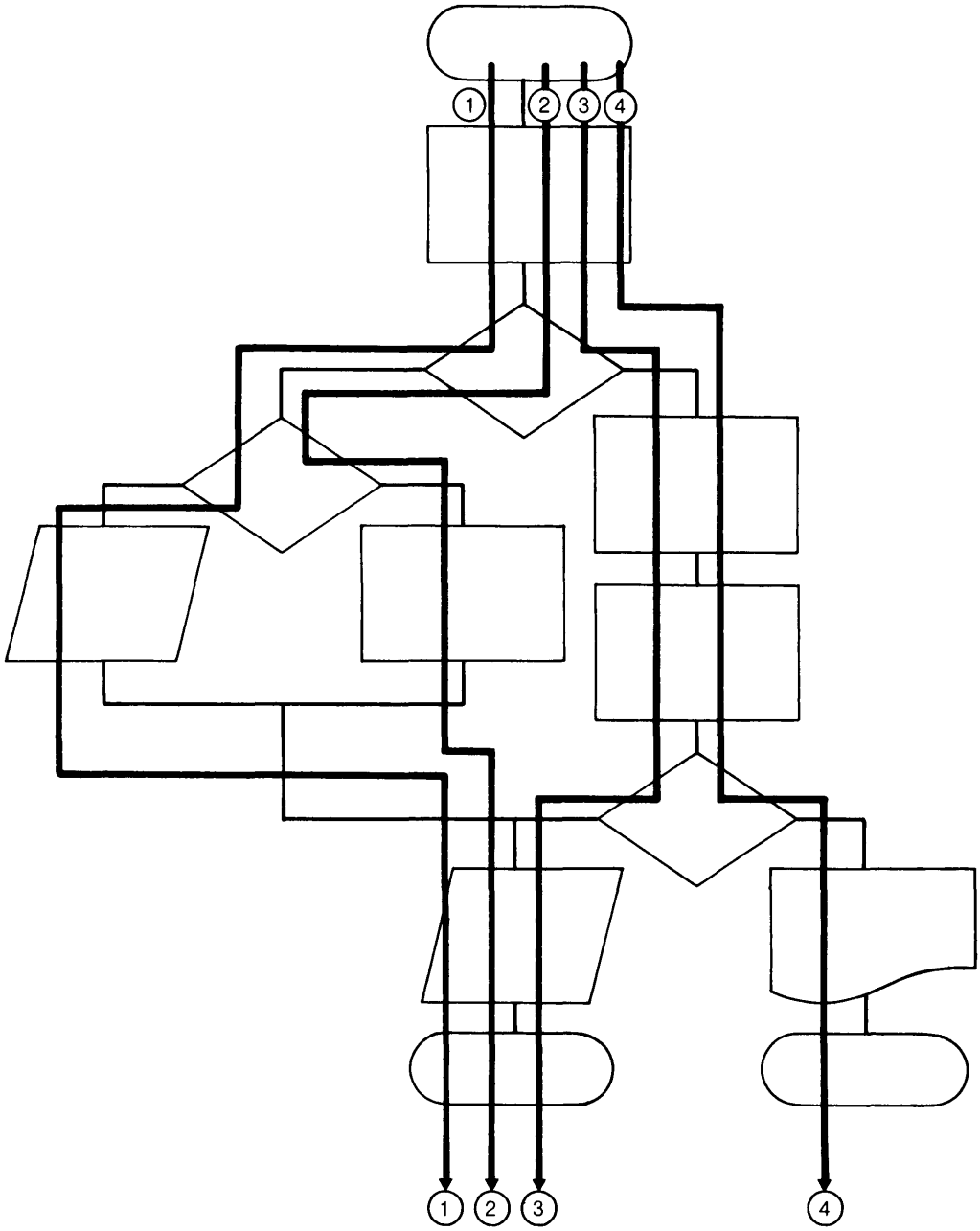
If the processing uses a history file, such as a master file, a test file must be prepared. This can be a copy of the file used for normal processing or a file especially prepared for the test. The auditor prepares the test data input and test files and computes the expected results of processing. Results obtained from processing the test data are compared with the precomputed results. If the two results are identical, the auditor has a basis for concluding that the program logic is consistent with the documentation. The comparison of the results also provides evidence of undocumented logic that is tested by the content of the test records.

When the test data concept was initially developed, test records were prepared on punched cards. As a result, test data are often called "test decks." In audit applications, the concept involved using test data to test a program prior to its use in normal processing and again at the end of the reporting period. In this manner, the auditor would be provided with some assurance that the same program logic had been used throughout the processing cycle. However, the use of test data to provide such assurance has three major difficulties. First, it is very difficult and time consuming to prepare and maintain a comprehensive set of test transactions that will adequately test a program. Second, testing a program at a specific point provides no evidence or assurance of the processing during the time period covered by the audit unless client library procedures can be relied upon to ensure that the program tested is the same as the program that was used during the period. Finally, programs change during the period because authorized maintenance activities correct errors and modify and enhance the processing logic.

Although a comprehensive set of test data is needed to test a program fully, audit testing may focus on sections of logic, such as input data validation.

It is desirable in either full or partial testing to develop the test data systematically. Three alternatives are a flowchart logic tracing method, a code logic tracing method, and a systematic error method.

- 1 *Flowchart logic tracing* Using, for example, a set of colored pencils, and beginning at the start of the program flowchart (or a section of the program flowchart in partial testing), the auditor traces every possible path through the program. Figure 14-3 illustrates this concept for four paths. When the paths have been identified, the auditor specifies data items that test each path, including the boundary value and boundary value plus and minus one.
- 2 *Code logic tracing* Using a copy of the program and colored pencils, the auditor marks all paths through the program listing. Each IF statement defines alternative paths. Just as in the flowchart logic tracing method, the auditor then specifies data items to test paths and boundary values.
- 3 *Systematic error method* The flowchart and code logic tracing methods are guided by the documentation. They establish test data to test logic as documented, however, they do not define any conditions that are not part of the documentation. An alternative or supplemental method, especially useful in testing of input data validation, is to generate error conditions that the program logic should handle in order to obtain evidence on the actual performance. For example, the systematic error method can test input record processing for the following:
  - Zeros where they are not allowed
  - Blanks where they are not allowed
  - Alphabets in a numeric field
  - Item size too small for the field
  - Item size too large for the field (in free-form input)
  - Limits on item values
  - Boundary value and boundary value plus or minus one
  - Very small values (to detect rounding problems)
  - Very large values (to detect overflow problems)
  - Negative values where they are allowed and where they are not allowed
  - Error records as first and last records (to test first- and last-time-through logic with error records)



**Figure 14-3** Flowchart tracing method for test data development

## TEST DATA GENERATORS

Test data techniques were developed primarily to assist programmers in the testing of application program logic, which requires the coding of a large number of data records. To reduce the difficulties in preparing test data records, a number of test data generators (TDGs) have been developed. These software packages are designed to simplify creation of test data records in virtually any format.

TDG programs use the file descriptions from the programs to be tested as the specifications for the file layout and file organization. By employing parameters, the user can describe the types of data to be generated for each of the data item fields within the test records. In addition, the user can specify the total number of test transactions or records to be created. Once these parameters have been supplied, the software package creates the required test records.

Figure 14-4 illustrates a set of parameters for generating a file of test data with a test data generator. The file is called INF082. It will contain twenty records called INVENTORY-TRANSACTION-RECORD. When generated, the file is to be in sequence according to the contents of the TRANS-PART-NUMBER field. Each record will contain four fields: TRANS-PART-NUMBER, TRANS-TYPE-CODE, TRANS-QTY, and TRANS-DATE. The first three data item fields are to be filled with random digits generated by the TDG. The first and third fields (TRANS-PART-NUMBER and TRANS-QTY) may use any of the digits from 1 to 9, but TRANS-TYPE-CODE may contain only the digits 1 and 2. The fourth field is to contain a constant value of 8X1231 in all records.

The output from the processing of the test data generator program is a file or files that can be used as input to the program or application system to be tested. Since the package will handle most of the clerical and technical burden required in the creation of test data, the task of generating such data is greatly simplified. Figure 14-5 illustrates the contents of the file created by the processing of the parameters from figure 14-4. An auditor who has access to a test data generator can create large volumes of test data in complex formats with very little effort. By reviewing the planned or typical contents of the fields that are to appear with the test records, the auditor can make decisions about the ranges and types of values to be generated in order to test the normal and exception processing for the application.

Some of the test generator packages have a feature that allows most data to be generated within the normal validation limits of the system and some records to be generated in excess of those limits. These randomly generated exceptions serve to test the handling of error conditions. Auditors will find this feature quite helpful.

## OPTIMIZER PACKAGES TO EVALUATE TEST DATA

In designing test data for large programs, it is difficult to analyze a program and construct test data that exercise all of its logic. Optimizer packages, originally developed for improving systems performance, can be used to evaluate the completeness of the test data in executing all branches of the code.

Optimizer packages are available from software vendors and other sources. The optimizer program monitors the execution of an application program and keeps a record of the lines of code within the program used during processing. At the end of the application run, the optimizer package prints out a list of all the lines of code within the program and, next to each line, indicates the number of times it was executed. Figure 14-6 illustrates a portion of an optimizer output report. In addition, some optimizer



0177	2	533965	8X1231
0365	1	758908	8X1231
0834	2	494054	8X1231
1227	1	049344	8X1231
1491	1	432833	8X1231
1750	1	046944	8X1231
2107	2	902751	8X1231
2458	1	079432	8X1231
2994	2	186640	8X1231
3559	1	806114	8X1231
4157	1	573834	8X1231
4548	1	941881	8X1231
4751	2	403657	8X1231
5114	1	309490	8X1231
5918	1	230029	8X1231
6180	1	311987	8X1231
8324	2	679244	8X1231
8559	2	695948	8X1231
8850	1	986248	8X1231
9827	1	463469	8X1231
RECORDS PRODUCED		=	20
TRANS-PART-NUMBER		=	83190
TRANS-TYPE-CODE		=	27
TRANS-QTY		=	9886772
TRANS-DATE		=	14824620

**Figure 14-5** Test data generator output

PROGRAM EXECUTION PROFILE		
05/08/8X		PAGE 07
STATEMENTS		EXECUTIONS
110010	PROCESS-TRANS-1 SECTION.	
110020	P0200.	
110030	ADD TRANS-QTY TO MASTER-QTY-ON-HAND.	12
110040	MOVE TRANS-DATE TO MASTER-LAST-RECEIVED-DATE.	12
110050	PERFORM WRITE-UPDATED-MASTER-FILE.	12
110060	IF DISC-ERROR-KEY EQUAL "Y"	12
110070	PERFORM DISC-OUTPUT-KEY-ERROR	0
110080	GO TO P0210.	0
110090	PERFORM UPDATE-CONTROL-TOTALS.	12
110100	P0210.	
110110	EXIT.	

**Figure 14-6** COBOL optimizer program execution profile

REPORT OF UNEXECUTED PARAGRAPHS		
05/08/8X		PAGE 36
300020	P2100.	PAGE 16.
300090	P2110.	PAGE 16.

**Figure 14-7** COBOL optimizer report of unexecuted paragraphs



packages produce a special report highlighting the lines or sections of code that were not used during processing. Figure 14-7 illustrates such a report

Lines of code with no usage following the processing of test data by the optimizer indicate that no test data records used the lines of code, either because the test data records were not complete or because the documentation used to define the test data did not include the untested logic. Note that the processing of a set of codes following a breakpoint does not mean that the boundary value itself was tested. If a boundary value of forty is used to branch into two different logic paths, test data of thirty and fifty will exercise the logic of the two paths but will not detect a coding error, in which forty-one is the boundary value.

If an auditor is able to determine the portions of the program that are not being tested, the test data can be expanded. By performing this procedure several times, the auditor can develop a set of comprehensive test data that fully exercises all parts of the program to be checked for audit purposes.

The optimizer can be used to evaluate test data created by hand or by a test data generator. The joint use of a TDG and an optimizer is a very powerful audit technique. This approach is particularly beneficial in the audit of complex systems.

## **Simulation or Parallel Processing to Test Application Processing**

Simulation, as an audit technique, is sometimes referred to as "parallel processing." It involves creating a model of the system to be tested. Rather than reviewing the actual system and attempting to substantiate its controls and processing results, the auditor reviews the application system to gain an understanding of what it is supposed to do and then uses a general purpose software package to create a model or simulation of the application processing.

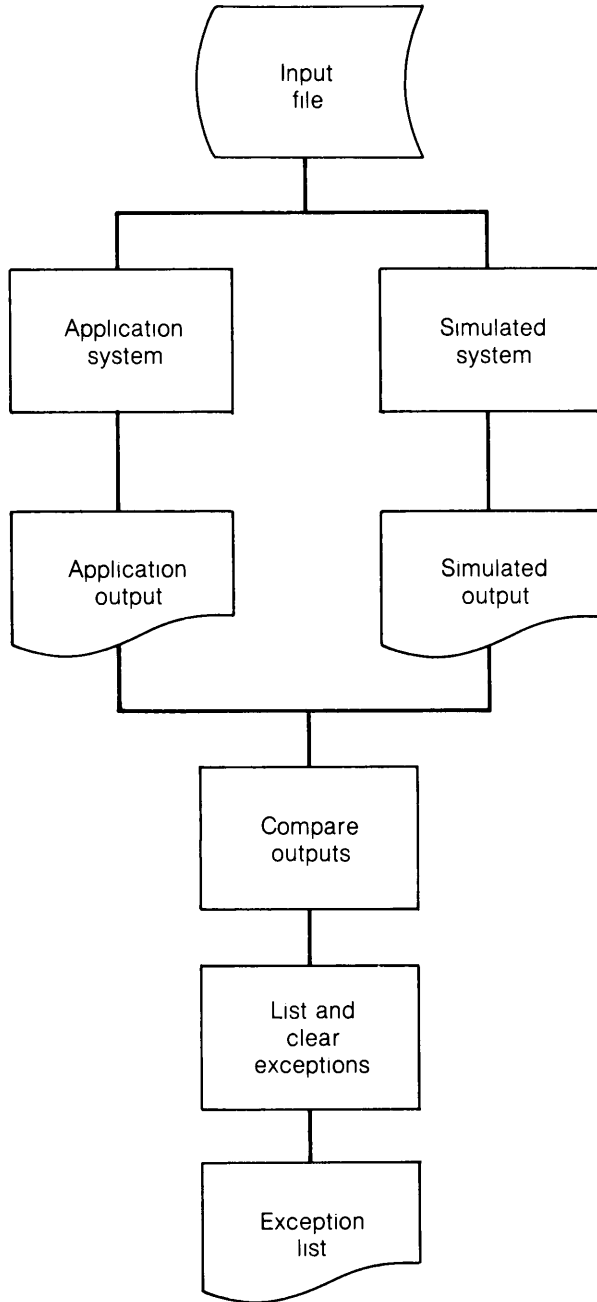
The auditor obtains actual transaction records that have been processed through the application system and processes these through the model of the application. The two sets of output are compared, and any differences are investigated to determine their causes. If all major differences are reconciled, the auditor has evidence that the system functions in accordance with his understanding of its operations. Figure 14-8 is a flowchart of a typical parallel processing operation.

The simulation technique has the advantage of focusing the audit testing without being tied to client program documentation; however, it presents the danger of overlooking important considerations. The testing provides evidence on processing that was performed rather than on programs purported to be used.

From a theoretical standpoint, the approach provides good audit evidence on application processing. It is limited by the difficulty of designing and implementing adequate simulation models for large systems. If an on-line accounting system required thirty man-years to develop and implement, it might take the auditor several months to create even a simplified model of the application system.

On the other hand, an auditor may choose to simulate only selected portions of the application processing procedures. Some auditors have used general purpose computer audit software packages to create models of relatively simple client systems or portions of more complex systems and to perform simulation tests. Auditors who have used this technique have reported useful and cost effective results.

As more powerful modeling software is developed, the use of simulation may increase. For the present, its use is likely to be restricted to relatively simple application systems or relatively small subsets of complex systems.



**Figure 14-8** Flowchart of parallel processing logic

## Summary

Auditors employ a variety of testing techniques in the audit of data processing systems. These techniques have been directed to the audit of both records and computer processing. While computers have made testing more difficult than it was in the past, auditors can use several approaches that will facilitate their testing efforts. This chapter explains tools and techniques for testing files, for static testing of application programs, and for simulation to test application processing.



---

# Computer Audit Tools and Techniques Used After Processing: Review of System Operations, Code Review, and Other Software Tools

# 15

The preceding chapter described three computer audit techniques that can be used after processing: testing of files, the static testing of programs, and simulation testing. This chapter covers additional techniques: review of system operations, review of program code, and other software tools.

## Review of System Operations

Review of system operations is an after-the-fact or passive test, using information generated as a by-product of normal processing. Generally this information comes from logs produced by the system or accounting data generated by processing.

### LOGGING FACILITIES

As part of their normal operations, most computer systems provide logs, which are files containing records of system activity. Logs of system activity provide capabilities for restart and recovery in the event of failure, facilitate file reconstruction, and keep records of communications traffic or transactions processed. As systems become more complex, the use of logging facilities increases.

System activity logs contain a great deal of information that can be useful to auditors. In on-line, real-time systems, logs often provide the only basis for establishing what occurred in the system at a particular time. As a result, auditors may wish to review the logging activity in the system to evaluate whether the logs will be useful in audit procedures.

Restart and recovery logs are often maintained in data communication systems or in complex multiprogramming systems to provide for recovery from a processing failure. These logs contain information that is useful to systems people; however, an auditor who needs to review system activity that occurred at a particular time should review these logs.

For example, in a fraud case that occurred in an on-line bank system, the only clue to how the bank records had been manipulated was provided in the information recorded on the restart and recovery log. Unfortunately, by the time investigators realized

## 236 Review of System Operations, Code Review, and Other Software Tools

that the restart and recovery log would supply them with the information they sought, the log tape had been scratched. The operations people made it a practice to scratch the restart and recovery log in forty-eight hours, unless some problem had occurred, since they regarded it as a tool that would be needed only in the event of a systems failure. If the logs may be needed for audit purposes, the auditors should review retention practices to make sure that they allow a reasonable time for subsequent audit review.

In some situations, the organization may decide to create a transaction audit log for audit purposes. In a simple format, this log provides a complete audit trail of all transactions processed through the system. A log record is written for every transaction that enters the system, indicating time of entry, source of entry, and complete transaction data. The log is maintained in date and time sequence order. An auditor uses this log to trace the history of a particular record or account over a period of time. The log's simple format is designed to be written on a tape drive without increasing overall processing time.

Although created for audit purposes in the company, the audit transaction log has proven useful to systems people as well. In a number of cases, systems people involved in attempts to debug systems failures have found the transaction audit log to be an invaluable aid in determining the status of the system at the time the failure occurred.

Some auditors encounter difficulty in using system logs in which the formats and layouts of the records are not designed to interface with the auditor's computer audit software or other data retrieval tools. It is difficult for the auditor to access and read records on the log without the use of utility software packages that are specifically designed to interface with the logs and prepare reports of recorded activity.

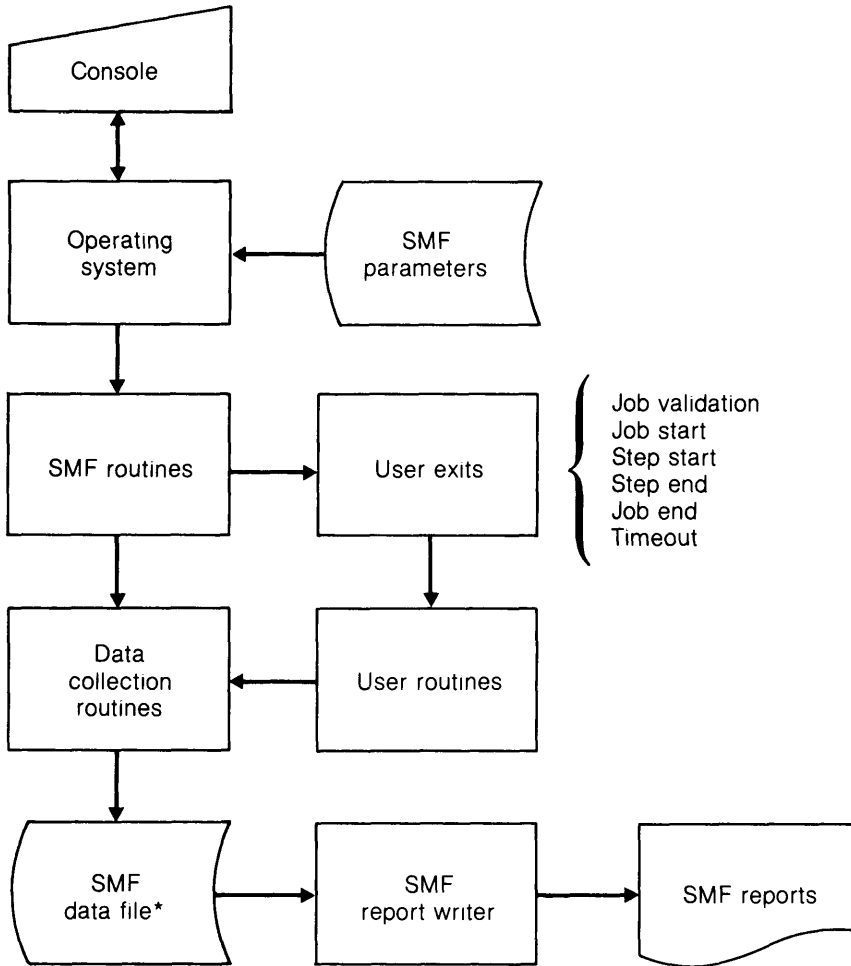
### **JOB ACCOUNTING FACILITIES**

Most computer systems provide a computer software facility for capturing data on the use of computer resources. The data provide the basis for an effective EDP cost accounting system and for a system to charge users for their share of the computer resources. The software for recording such data has been available for several years, but it has not been widely used by auditors. Some auditors are simply not aware that the software is available, others do not possess the expertise to understand and use the accounting data in an audit.

Perhaps the best known of these accounting systems is System Management Facilities (SMF), an IBM product. However, almost all the major vendors have implemented similar software packages for their equipment. Figure 15-1 shows the processing flow in an SMF application, and figure 15-2 shows a representative report derived from the accounting data.

The computer system accounting data collection is not designed for audit use, but there are situations in which the system accounting data can be used in audit procedures. The examples below include activities associated with both financial audits and management or operational audits.

- The auditor can scan system accounting records to determine that applications were properly authorized, were run at the scheduled time of day, and were run with anticipated frequency within a given period. This type of review will help the auditor to evaluate control over system usage. (See figure 15-3)
- The auditor can sort system accounting records into user or account number order.



Job validation  
 Job start  
 Step start  
 Step end  
 Job end  
 Timeout

\*Could be on tape

**Figure 15-1** SMF processing flowchart

05/08/8X															PAGE	02
DATE	TYPE	JOB NAME	ENTER HH MM	START HH MM	DONE HH MM	PRINT LINES	TOTAL I/O'S	TASK TIME HH MM SS	REAL TIME HH MM SS	CORE REQD	CORE USED	END CODE				
8X128	T	COB0LL	13 08	13 11	13 27	404	1849	00 00 37	00 01 27	30K	30K	000				
8X128	T	COB0LL	13 09	13 11	13 18	126	1132	00 00 15	00 00 58	30K	30K	000				
8X128	P	INPO81	13 12	13 12	13 29	293	22003	00 02 24	00 11 26	42K	31K	000				
8X128	P	INPO82	13 13	13 30	13 52	58	32493	00 17 14	00 23 11	60K	60K	000				
8X128	P	INPO83	13 58	13 58	14 09	3	21333	00 07 36	00 11 09	28K	28K	099				

Figure 15-2 SMF reporting system



PROGRAM EXECUTION REPORT

DATE: 1/20/6X  
TIME: 08:19

FROM 07:53 01/19/6X  
THRU 08:07 01/20/6X

PROGRAM ID	CHARGE NUMBER	TOTAL TIME DAILY	TOTAL TIME MONTHLY	DAILY EXECUTES	MONTHLY EXECUTES	DIRECT TIME DAILY	DIRECT TIME MONTHLY
DISPKV	000001		11:48		1		:05
DPPALL			1:41:00		17		1:10:34
DSKOUT			1:00		3		:16
LCADMP			9:56		6		1:10
PACKUP			24:38:43		22		24:02:43
PEDOUT			3:10		3		:07
*TOTAL*	000001		26:45:37		52		25:14:55
CPADCH	000009		:00		1		:00
DPPALL			5:09		17		1:52
DSKOOT			:42		1		:05
HAMMER			:03		1		:03
LCCNTL			7:12:45		10		2:53
LGAD01		3:35:00	1:51	1	2	:58	:44
LGAD02		8:26	9:36	1	2	2:35	2:45
LGAD03		:31	:47	1	2	:06	:09
LGAD04		9:05	9:23	1	2	1:09	1:12
LGAD05		:35	:37	1	2	:10	:10
LGAMMT		:08	:08	1	1	:01	:01
LOADER			:36		2		:00
LOADMP		1:31	14:46	4	11	:38	5:04
LOGCON		1:20	1:36	1	2	:58	:58
LOGOUT		1:08	1:22	1	2	:12	:12
MPSUTL			4:05		1		:20
PACKUP			6:29		2		:53
PEDOUT			:52		10	:01	:07
*TOTAL*	000009	3:59:34	8:10:47	3	71	7:28	17:28

Figure 15-3 Program execution report

## 240 Review of System Operations, Code Review, and Other Software Tools

to test user cost allocations. This procedure may be particularly useful in auditing job accounting systems.

- Job accounting data captured by the system can be compared with the hardware vendor's billings to detect overcharges or billing errors.
- The auditor can review information on the use of an input/output device to determine whether there is excess equipment. (See figure 15-4.)
- System accounting records can be sorted by job number and by the time required to process the jobs. This data can then be compared with the volume of records or transactions processed to highlight specific processing problems and to indicate inefficiencies in system operation. Unusual variations in running time for a particular job also may be caused by unauthorized manipulation of the system.

In order to perform these analyses, the system accounting data items must be organized into a useful form. This processing can be done by using commercially available software for processing system accounting data or by writing a specific program. Figure 15-5 illustrates the audit processing of SMF data using an extract program to select data items and a report writer to summarize and print the data.

### Review of Program Code

Although a detailed review of the program instructions (the code) is the most effective way to gain an understanding of the processing that takes place within a program, code reviews are used in audits only when other alternatives are ineffective. Code reviews are very time consuming, and, since programs undergo frequent changes and corrections, previous code reviews must be updated at each audit. Also, they require a fairly high level of EDP expertise, including a reasonable amount of training and experience in computer programming. Other techniques do exist for obtaining evidence about program logic, and they are generally more cost effective. Finally, for a code review to be practical and worthwhile, client program documentation must be adequate, and program library procedures must provide reasonable assurance that the program being reviewed is the program version in use.

Code reviews can be useful in certain situations, and an auditor should understand the technique. In some cases, the auditor can perform code reviews to become familiar with client programming and change procedures.

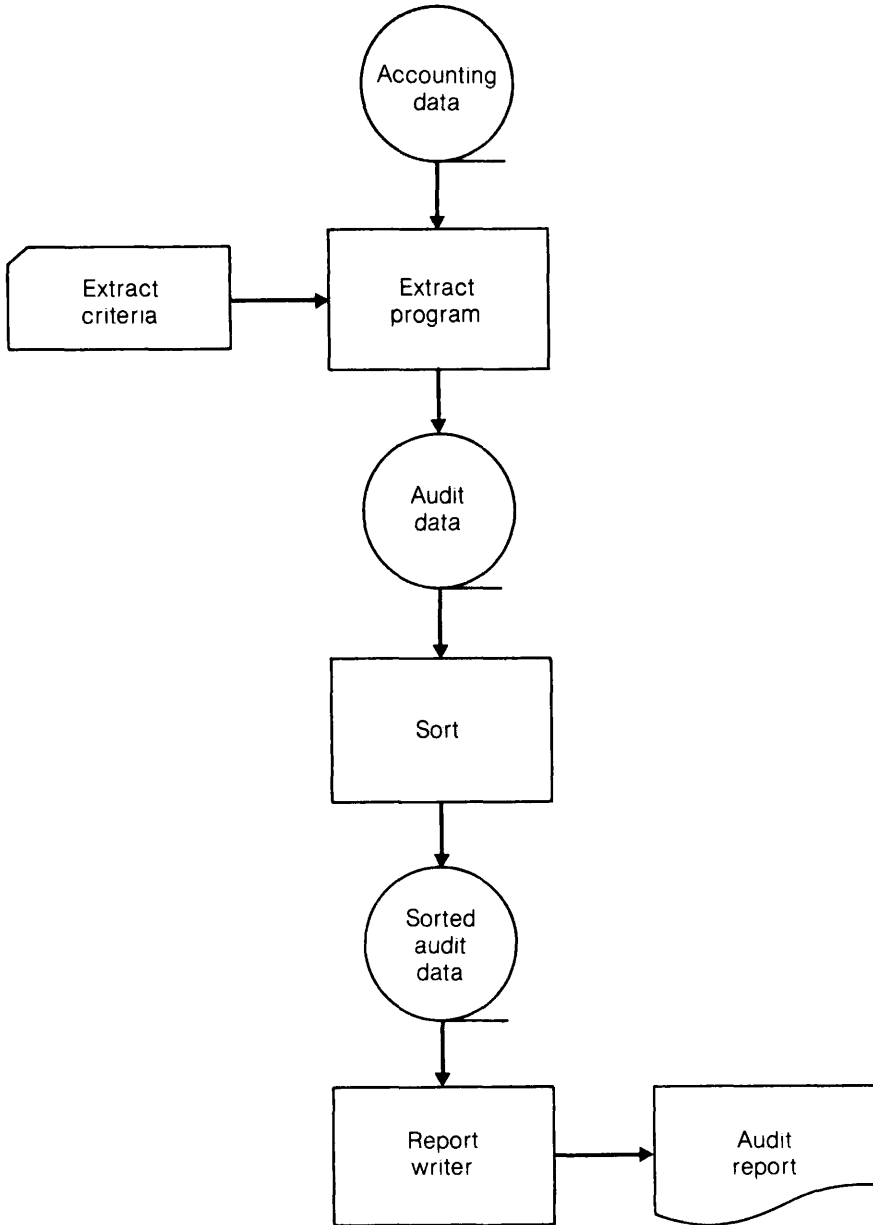
Generally, the use of the technique has been restricted to highly sensitive applications that have a particular control or financial significance. For example, in one case, auditors reviewed the program code in the edit and update routines for a large-scale, on-line database system. These routines were examined because they were key elements in the overall structure of control for the system.

### AN OUTLINE OF THE CODE REVIEW PROCESS

Because the code review process is complex, a key factor is the establishment of an effective code review plan. Before undertaking a large-scale code review, it is a good idea for an auditor to gain experience by performing a review of a smaller program or application. The outline provided here describes the typical plan of a review of a COBOL program, but the activities described apply to virtually any programming language.

DATE: 01/20/8X TIME: 8:20		PERIPHERAL USAGE REPORT					
		FROM 7:53 01/19/8X THRU 8:07 01/20/8X					
		FILE TIME	RECORDS	BLOCKS	HARDWARE INPUT	EXCEPTIONS OUTPUT	I-O
00/1	DSCUDO-RD	1:17:23	28,567	28,567			
02/0	DISK	6:38:44	249,501	67,680			
03/1	TAPE	3:55:30	301,842	63,272	2	44	
03/2	TAPE	2:37:34	582,028	164,452	12	24	
03/3	TAPE	2:52:25	130,507	44,901	168	18	
03/4	TAPE	36:20	191,503	36,132	2	5	
04/1	TAPE	2:46:04	354,437	147,301	2	5	
07/0	PRINTER	5:54:15	146,119	146,119			
08/0	PRINTER	7:17:59	259,029	259,829		4	
09/1	DISK PACK	14:33:56	236,328	143,806	35	1	
09/1	FRT-BACK	1:07:03	12,231	12,231			
MONTH-TO-DATE TOTALS.....							
PROCESSOR TIME		64:47:41					
TOTAL IDLE		35:36:12					
RERUN TIME		48:41:55					
HALT-LOADS		22					
EXECUTES		814					
DISK		37:36:04					
00/1	PSEUDO-RD	3:54:34					
	FRT-BACK	:30					
03/1	TAPE	8:27:08					
03/2	TAPE	8:10:22					
03/2	FRT-BACK	3:10					
03/3	TAPE	10:30:56					
03/4	TAPE	2:45:57					
04/1	TAPE	8:13:30					
04/1	FRT-BACK	2:40					
06/0	READER	18:24:07					
07/0	PRINTER	12:48:09					
08/0	PRINTER	16:31:39					
09/1	DISK PACK	39:36:44					
	FRT-BACK	3:48:30					

Figure 15-4 Peripheral usage report



**Figure 15-5** Audit processing of job accounting data

- 1 Review the EDP standards within the organization. In many cases, these standards provide the auditor with an overview of the approach taken in writing and documenting computer programs.
- 2 Identify the audit objectives for the code review. The audit objectives focus on program logic having control and financial significance, in contrast, code reviews for efficiency of program design focus on storage use, running time, and so on.
- 3 Obtain a listing of the source code for the program or set of programs to be reviewed. The auditor should determine that the list reflects the current status of the program, this can be done by reviewing program change control or librarian procedures. If the auditor doubts that the listing is current, he might ask the operating people to recompile the code and to use the program in the next production run under audit control. If they are willing to follow this procedure, then the source code version is probably current.
- 4 Obtain a copy of the documentation in support of the program. If the documentation is inadequate, the auditor might consider asking the data processing people to prepare better documentation. If this is not possible and audit review of the program code is considered essential, it may be necessary for the auditor to document the purpose and logic of the program. Note that program logic flowcharts frequently are not used for documentation if a high-level language is used for coding, since the high-level code should provide sufficient logic documentation.
- 5 When the supporting material has been assembled, the auditor should begin the program analysis by determining the input and output files used by the program. If the auditor is looking at a COBOL program, a review of the SELECT clauses identifies the files that have been defined and the hardware devices they were assigned (tape, disk, and so on).
- 6 Compare the details in the data division of the COBOL program with the details of the documentation. In particular, the auditor should test that the fields and field sizes specified in the data division correspond to those in the documentation. This is an extremely important step because a change in a program for an unauthorized manipulation often involves a special field or identifier within the records being processed. The auditor should be particularly alert for any differences between the definition of the records in the program code and the supporting documentation.
- 7 Note the presence of the COBOL REDEFINES clause. The REDEFINES clause is a powerful instruction in a COBOL program that can be used to redefine a record format into an alternate form. As a result, a program might be subject to manipulation through the use of a record format created by a redefinition of a legitimate record or field. Therefore, the auditor should determine the validity of any REDEFINES used in a program.
- 8 Request a printout of several records from each of the files. In this way, the auditor can employ his understanding of the file content obtained from the documentation and the program code to judge the contents of the file printouts. Any differences between what the auditor considers the correct contents of the printout and what is actually contained in the printout should be investigated, since they may indicate processing errors or attempts at unauthorized manipulation. Any discrepancies should be discussed with the data processing personnel and resolved to the auditor's satisfaction.
- 9 Review the OPEN and CLOSE commands in the COBOL program. Normally, there should be one OPEN and one CLOSE statement for each file identified in the SELECT clauses for the program, and any exceptions to this rule would be rare. If

## 244 Review of System Operations, Code Review, and Other Software Tools

more than one OPEN or CLOSE appears for a particular file, the auditor should know the reason why. The auditor should also be alert to any variations between the names of files defined in the SELECT clauses and those identified in the OPEN and CLOSE statements.

- 10 Scan the program for the presence of IF statements. These statements control the flow of the processing logic within the program; therefore, the auditor should be able to verify why each IF statement is used. Any extraneous or unusual IF statements should be thoroughly investigated.
- 11 Review the use of the various forms of the PERFORM and GO TO commands within the program. These commands are used to alter the flow of logic within a program. The auditor should be able to understand why these commands are used and should be able to follow their impact on the flow of logic in a program.
- 12 Review the use of the verbs that are used to pass parameters to subprograms. Examples are CALL, ENTER, and RETURN. Each of these commands involves an interaction between the program being reviewed and some other program. To understand the processing, the auditor may have to review the programs being called or entered.
- 13 Review the paragraphs and sections of the program that are directly related to the audit objectives. For example, the data validation sections may be reviewed for completeness and compliance with program specifications.

### SOFTWARE REVIEW AIDS

A review of the program code can be a time-consuming and difficult task, since a program of modest size can contain five hundred to one thousand lines of coding. Software aids can be used to reduce the clerical detail associated with the code review.

#### Flowcharting Software

Flowcharting software (discussed in chapter 5) may prove very useful in analyzing a program. If the program is not coded in a readable form or has not been well documented, the auditor might use a flowchart package to diagram the logic of the program. (See chapter 5, figure 5-5) A graph of the logical flow often provides a better understanding of a program than a review of the source code listing can provide.

#### Cross-Reference Systems

It is very difficult to pick out all the uses of a particular data name or verb (command) in a program. A cross-reference system can simplify this task. It identifies the point in a program where each data name is defined and lists all of its subsequent uses, which facilitates the tracing of activity that affects a particular field or uses a specific COBOL verb (figure 5-6). Also, a GO TO or PERFORM verb may cause the program logic to shift to some other paragraph or section within the program. This point is identified by a paragraph or section name, and it may be difficult to locate that name in the program listing. The cross-reference listing will identify the location of the name being referenced.

## Record Layout Software

The key to understanding any program is a knowledge of the input and output files. The auditor needs to have adequate documentation describing the record layouts and content. In the absence of formal documentation, the auditor might consider the use of record layout software (discussed in chapter 5) to prepare a description of the files (See chapter 5, figure 5-7)

## Optimizers

Software optimizers (discussed in chapter 14) can be useful in analyzing program logic. The auditor may be interested in reviewing any section of the program that was not used during normal processing, and it is very difficult to detect, on a manual basis, a program code that is not normally executed. If the actual application program has been executed under the control of a software optimizer, the auditor will be able to obtain a list of unused program code, which may indicate inefficient program design or unauthorized code.

## AUDIT CONTROL OF PROGRAM CHANGES

If a program that was subjected to code review is likely to be of concern to the auditor at some future time, the auditor also should evaluate organizational controls over program modifications and establish audit procedures to detect future undocumented changes to the program. This action may help the auditor to avoid having to repeat the entire code review at a subsequent date. Library software facilities can supplement change control and provide an audit trail of changes that have been made.

## Change Detection

Three change detection techniques that might be used by the auditor are the creation of a control copy of the program, an instruction count, and a program code hash total.

To create a control copy of the program, the program source code is copied onto magnetic tape and retained under audit control. The auditor can, at a later time, compare the current version of the program with the control copy, using the source code compare facilities that are available as part of some vendor software libraries. The auditor need review only those portions of the code that were changed. If the changes appear legitimate and were properly approved, the auditor has a basis for accepting the revised version of the program.

A count of the instructions within a program can also be used to check for unauthorized changes. For example, if a program contained 560 instructions when it was last reviewed and the current version of the program still contains 560 instructions, there is some basis for deciding that no changes have been made. This assumption is dubious, however, because someone can easily modify a program and maintain the same instruction count that existed before the revision.

Hash totals provide a stronger control. Each line of code is treated as a field of data, these fields are added together, and the resulting total is called a hash total. It would be quite difficult for someone to make a program change without disturbing this hash total. This approach provides an effective starting point for determining whether

## 246 Review of System Operations, Code Review, and Other Software Tools

or not a program has been changed, although the auditor must then identify the specific changes. Unless the changes can be isolated, the entire review process may have to be repeated.

### **Librarian Software**

Librarian software is used to provide control and to facilitate the maintenance of a software library. These functions require the monitoring of all changes. Most librarian packages produce a log of all changes, and some of them maintain copies of the prior version of the program as part of an archival storage system. Further, as part of its log, some librarians maintain a record of the date of the last update to each program (see chapter 5, figure 5-10).

By comparing the date of the audit control version of the program with the most recent revision date on the log, the auditor can determine whether the program has been revised since the last review. If the date of the most recent modification is subsequent to the auditor's review, the log entry for that date shows the changes that occurred. The change log also provides the date of the next prior change. If that change is also subsequent to the date of the auditor's review, the program must be reviewed. By tracing along the trail provided by the change log, the auditor can identify the details of all changes that have been made since the program was reviewed.

Most librarian software packages only maintain control over source code, however, a few extend this same control to object code. Since the object code is the actual language used to perform the processing, control at this level is also needed. Working with an object code librarian system, an auditor can review the logs to determine that the program under review has not been subject to changes that were applied directly against the object code.

### **THIRD-PARTY CODE REVIEW**

As technological developments decrease the cost of computer hardware, the cost of developing and maintaining software is increasing proportionally. Many data processing installations are currently spending more money on software than they spend on hardware. To some extent, this trend occurred because most organizations preferred to do their own development work rather than make use of available software. However, this attitude has changed. The high cost of preparing and maintaining software has caused many installations to use application program packages.

As the use of standard application program packages becomes more widespread, there may be an economic need for a central audit review of the packages. If every auditor whose client uses a package is forced to review its logic and evaluate its controls, a considerable duplication of audit effort will result. Further, such reviews are likely to place severe demands on the resources of software vendors. Therefore, a third-party approach to the review of application program packages may be appropriate.

In the third-party approach, an auditor would be engaged by the vendor of the package or by a user group to review the application package and its controls and to express an opinion on them. Further, the auditor would supply other audit groups with a controlled copy of the program code. An auditor of a client who uses the application package would rely partly on the work done by the third-party auditor in reviewing the package and its controls. The auditor could further evaluate the appropriate level of this reliance by comparing the version of the program being used by the client with the copy provided by the third-party auditor, any discrepancies between these two ver-



sions of the program would probably indicate that the client has modified the package. The auditor would have to review the changes to determine their impact on the controls in the application program. Chapter 18 contains further discussion of third-party reviews.

## Other Software Tools

In addition to the program packages or systems that have been discussed already, there are other types of software that can be useful to auditors. Therefore, an auditor should carefully review software sources to determine if available packages satisfy audit needs. Sources of software are vendor libraries, software catalogs, and time-sharing services.

### VENDOR LIBRARIES

All major hardware vendors have extensive libraries of software in support of their equipment. Usually, these libraries contain many different types of programs. Some are programs the vendor has produced as part of a product line; others may have been written by vendor field personnel. Even though programs written by field personnel are not considered fully supported products, they are available from the vendor. Finally, the library may also include programs written by users and distributed by the vendor.

Many of the software offerings that appear in such a library are application or system oriented, but some of them can serve audit purposes. For example, a review of the catalogs published by two major hardware vendors identified the following packages:

- A flexible and reasonably priced test data generator that creates a variety of files containing records with field contents based on parameters supplied by the user.
- An interactive query and report processing system that can be used by a nontechnician who wants to obtain data from a database file. The data may be displayed visually or printed in an output report. The entire system operates interactively. If the user is not satisfied with the results, instructions can be revised immediately, and the job can be rerun.
- A batch query facility that provides an interrogation and reporting capability on a batch basis. While not as flexible as the interactive query and report processing package, this system does have the ability to read a file and to generate printed reports based on user parameters. It provides a very simple way to obtain information from a computer file.
- A graph analysis program that uses job accounting data as input. Based on this data, the program constructs a series of graphs that provide a clear, pictorial representation of system usage and activity, which provides a comprehensive overview of EDP operations.
- An audit source code compare utility that matches two versions of a program and prints a listing that analyzes any changes or modifications. This package provides logic that allows a sophisticated comparison of the two program versions. It highlights and identifies differences caused by additions, deletions, and changes. If an auditor reviewed a program and wanted to review subsequent changes, this source code compare program could be extremely useful.
- A programming control facility that provides a very high level of control over all the time-sharing activities in the system. For example, it is possible to restrict the use of

## 248 Review of System Operations, Code Review, and Other Software Tools

sensitive time-sharing commands to selected users of the system, which represents a significant improvement in control

- An analysis program that reads the log tape produced by a database, analyzes its content, and converts data to a simple format that can be easily manipulated. In its normal state, the database log is written in a very complex format. The log tape analysis program reads the log tape and generates information in a format useful to the auditor
- A dump utility program that retrieves and prints the contents of virtually any file in the system. For this reason alone, it is of interest to auditors. It can give them many of the capabilities of general purpose audit software. However, perhaps of even more concern to the auditor is the fact that this utility program provides a general purpose capability to modify the contents of files without having these modifications subject to any of the controls provided by the system. From the audit standpoint, it represents a very serious potential control threat
- A statistical analysis package that analyzes the contents of files and prepares histograms that describe their population. This package can be extremely useful to the auditor who is contemplating the use of statistical sampling. It provides much of the information that is needed to construct a sampling procedure
- A program for generating random numbers and arranging them in a variety of sequences specified by the user. This package will generate numbers in almost any form that might be required for use in an audit situation

Because these products were not designed for the auditor, their audit uses are not always readily apparent. The following situation illustrates the need to translate audit needs into data processing functions in order to identify software packages. In reviewing the financial records of a large firm, the auditors were concerned because they had no way to detect if an operating system had been subject to unauthorized modification. They discussed this problem with representatives from major hardware vendors, but the vendors could not provide a solution. One day one of the auditors mentioned this problem in a conversation with a systems programmer, who informed him that a standard utility program, available as part of the operating system, had been used for years to detect any unauthorized changes. This utility had been implemented by the vendor as a tool for troubleshooting a system to help determine if the problem had been caused by the client's programming staff. The utility was well documented and had been used by system engineers for several years. This utility provided exactly the data needed by the auditor.

### **SOFTWARE CATALOGS**

Other sources of software for audit use are the reference services, catalogs, and program exchange listings that are published on a regular basis. Reference and program exchange services provide their subscribers with listings of available programs, usually categorized by type of program and by required hardware. An auditor with a particular requirement should review these potential sources of help before deciding to write a program. No list of sources is provided here because it is difficult to list all available sources and companies in this field change often.

### **TIME-SHARING SERVICES**

A number of commercial time-sharing services maintain libraries of programs specially designed to assist auditors. These libraries are available on an interactive basis to

anyone who has a remote terminal and has contracted for the use of the time-sharing services. Users of the service are charged for the time and resources they actually use.

For the most part, time-sharing programs handle applications that involve very little input but have heavy computational requirements. The following are some typical applications:

- Mortgage amortization
- Rule-of-78 loan schedules
- Accounting and disclosure information for lessees and lessors (SFAS 13)
- Imputed interest (APB 21)
- Earnings per share (APB 15)
- Depreciation calculation and analysis
- Inventory analysis
- Present value computations
- Financial ratio trends and analysis
- Balance sheet analysis
- Bond interest calculations
- Real estate project evaluation
- Investment analysis
- Statistical sampling
- Correlation analysis
- Regression analysis

Some examples of these applications (figures 15-6, 15-7, and 15-8 at the end of this chapter) should illustrate the kind of processing provided by time-sharing. Figure 15-6 shows the processing of a program to compute financial ratios. The auditor supplies some basic financial data in response to questions from the system. The data are used to compute the thirteen common financial ratios.

Figures 15-7 and 15-8 illustrate random number generation applications. In figure 15-7 the auditor wants to select ten days at random throughout the entire year, excluding Saturdays and Sundays, the system comes back with a list of ten random dates. For some applications, the auditor may want to make a random selection of transactions or items from a computer printout, a subsidiary ledger, or some similar source. Figure 15-8 shows the use of a time-sharing program that will generate random page and line numbers. The auditor has decided to select fifteen items from a 356-page list that contains sixty items per page, the time-sharing system will make the required selection and print out the fifteen items in numerical sequence.

## Summary

Many software aids are available for audit procedures with EDP systems. Over the years, there have been very few software developments that could be identified as unique EDP audit tools or techniques. It seems likely that this situation will continue because almost all functions needed by auditors are also needed by data processing personnel. Therefore, auditors should seriously investigate available software.

CPA > PRATIO	
CLIENT'S NAME	: ABC COMPANY
DATE	: DEC 31 198X
CURRENT ASSETS	: 16600
CURRENT LIABILITIES	: 12000
NET INCOME	: 1800
SALES	: 27000
NET WORTH	: 8600
FIXED ASSETS	: 2000
TOTAL DEBT	: 0
NOTES & ACCOUNTS RECEIVABLE	: 2300
INVENTORY	: 1800
FUNDED DEBT	: 0
ABC COMPANY	
DEC 31 198X	
<u>PERTINENT RATIOS</u>	
CURRENT RATIO	1.38 TIMES
NET INCOME ON NET SALES	6.67 %
NET INCOME ON NET WORTH	20.93 %
NET INCOME ON NET WORKING CAPITAL	39.13 %
NET SALES TO NET WORTH	3.14 TIMES
NET SALES TO NET WORKING CAPITAL	5.87 TIMES
COLLECTION PERIOD	31.09 DAYS
NET SALES TO INVENTORY	15.00 TIMES
FIXED ASSETS TO NET WORTH	23.26 %
CURRENT DEBT TO NET WORTH	139.53 %
TOTAL DEBT TO NET WORTH	0.00 %
INVENTORY TO NET WORKING CAPITAL	39.13 %
CURRENT DEBT TO INVENTORY	666.67 %

**Figure 15-6** Financial ratios from time-sharing service

```

HOW MANY SAMPLES, MONTHS, LOCATIONS? 10,12,1
EXCLUDE WHICH DAYS:
      2 = SATURDAYS AND SUNDAYS
      1 = SUNDAYS ONLY
      0 = NO EXCLUDED DAYS ? 2
ENTER THE MONTH, DAY, YEAR OF THE FIRST SUNDAY IN THE PERIOD? 1,1,8X
THANK YOU

      MONTH DAY
      3     1
      3     9
      4     3
      4    17
      8    23
     10     2
     10     5
     11    15
     11    30
     12    28

RANDOM NUMBERS PRINTED - - - - 10
TOTAL POPULATION - - - 260

```

**Figure 15-7** Random number generation to produce random days for audit sampling

HOW MANY SAMPLES, PAGES, LINES PER PAGE? 15,356,60

THANK YOU

PAGE & LINE

10 45

30 17

32 49

73 46

77 8

81 54

88 55

120 10

153 31

158 31

158 54

246 18

263 16

265 1

280 22

335 40

RANDOM NUMBERS PRINTED ----15

TOTAL POPULATION --- 21360

**Figure 15-8** Random number generation to produce random page and line numbers for audit sampling

---

# Computer Audit Techniques Used Concurrently With Processing

# 16

Audit techniques used concurrently with processing, called "dynamic testing," permit the auditor to conduct audit testing during the normal operation of the application system. Tests conducted in a live environment provide useful audit evidence on processing logic and processing procedures. However, concurrent testing has inherent dangers. For example, some auditors who have attempted dynamic tests have modified actual transaction data and, as a result, distorted financial results or caused operational problems. In one case involving a military system, audit tests changed inventory quantities on a master file, causing the reorder of materials that were actually on hand. Therefore, the concurrent techniques should be evaluated in terms of both the audit evidence to be obtained and the possibility of interference with normal data processing.

A number of techniques allow selection or testing to be performed concurrently with operations. These have a variety of names and acronyms, some of which are variations of others. The term *tagging*, as used in this chapter, is a generic name applied to methods for marking, through a special code, records that are of interest. This chapter describes eight techniques used for concurrent testing, which can be divided into four major types of testing.

<u>Type of Testing</u>	<u>Technique</u>
1 Testing of programs concurrently with normal processing	Integrated test facility (ITF)
2 Transaction selection for audit examination based on— <ul style="list-style-type: none"><li>● Programmed criteria</li><li>● Audit criteria inserted at application execution</li></ul>	Systems control audit review file (SCARF) On-line audit monitor
3 Transaction selection and process data recording <ul style="list-style-type: none"><li>● Process data on tagged transactions written on the audit file</li></ul>	Audit indicator (AI)

(continued)

## 254 Computer Audit Techniques Used Concurrently With Processing

<u>Type of Testing (cont )</u>	<u>Technique (cont )</u>
<ul style="list-style-type: none"><li>● Process data on tagged transaction written as part of the transaction record</li></ul>	Extended record
<ul style="list-style-type: none"><li>● System status information on the tagged, selected transaction</li></ul>	Snapshot
4 Audit processing facilities included in application programs	
<ul style="list-style-type: none"><li>● Written as part of application</li><li>● Attached to application</li></ul>	Audit modules Audit hooks

The concurrent techniques can be used by either independent or internal auditors; however, the evidence obtained and the conditions for effective use suggest that they are generally more suitable in an internal audit environment. The evidence obtained is primarily related to current system performance, and the conditions for effective use are related to the auditor's familiarity with the details of the processing logic for the application being tested, his technical competence to supervise implementation and use of the technique, and his availability for on-site audit supervision.

The results of a questionnaire survey of forty-five internal auditors and fifteen independent auditors with experience in EDP audit techniques indicated agreement that the concurrent techniques are mainly for internal audit use.<sup>1</sup> However, an independent auditor should be familiar with concurrent techniques in order to evaluate the work of internal auditors who use them. Even though the concurrent techniques were used infrequently, those who had used them generally considered them to be effective.

### Integrated Test Facility (ITF)

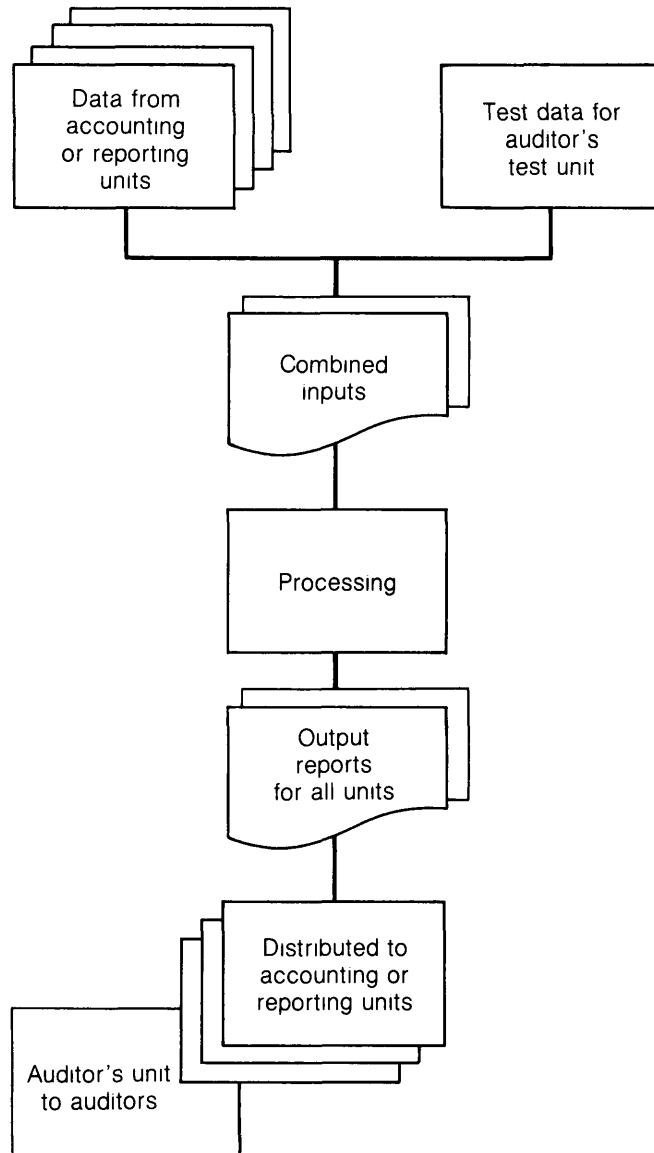
The integrated test facility is used for testing an application system with test data during normal operation. It can be used for applications that perform the same processing for different subunits but produce a separate set of output for each subunit, based on a subunit code in the input data. For example, the application program processes data for separate branches, separate departments, or separate subsidiary companies using the same program, but it keeps records separate based on the code that identifies the subentity. If there are four branches, a data item in each input record indicates the branch to which it applies. For example, values for the data item might be 1, 2, 3, or 4.

In the integrated test facility a fictitious category is added to the system and assigned a subunit code. That entity is an auditor-created branch, department, or company that is under the auditor's control (figure 16-1). The auditor submits test transactions for the audit entity as input to the system. These transactions are coded with the audit entity code so they can be processed for the entity that the auditor created. For example, if a system is processing sales transactions by branch, the auditor creates an audit sales branch and assigns it a code. The auditor creates test transactions and enters them into the normal processing input. The audit branch sales transactions are coded with the audit branch code so they can be posted to the records of the fictitious branch.

---

<sup>1</sup> Gary L. Tobison and Gordon B. Davis, "Actual Use and Perceived Utility of EDP Auditing Techniques," *EDP Auditor* (Spring 1981) 1-22





**Figure 16-1** Integrated test facility

The output for the audit entity is in the same form as output for other entities. If the results of all subunits being processed are also combined for higher-level reports and financial accounting, the results of the audit entity must be excluded from the summarized process or removed by subsequent procedures.

Thus, using an ITF, the auditor can develop a body of test data and have it processed as part of the normal transaction flow. Since the test transactions affect only the audit entity, the auditor is able to easily identify the results of the test processing. ITF results provide evidence, obtained under normal operating conditions, of logic in the application program in actual use and provide the auditor with evidence on operating procedures and error handling.

The ITF technique has some attractive features. Little effort is required to create the fictitious entity if the application is already designed for multiple entities based on a code; if it is not, the effort and cost will be greater. It is important to obtain high-level management approval and support for this approach, although the presence of the fictitious entity should be kept confidential. If it becomes known that a particular branch actually represents the auditors, the effectiveness of the ITF may be compromised. A programmer who can identify the audit entity can insert coding to change or bypass some of the normal logic whenever a transaction for the audit entity is processed.

## Transaction Selection

The transaction selection methods select transactions for audit examination and evaluation. Two examples of transaction selection methods are the system control audit review file (SCARF) and the on-line audit monitor. They differ in the transaction selection criteria provided to the program.

### SYSTEMS CONTROL AUDIT REVIEW FILE

The SCARF technique uses audit routines imbedded within an application program to monitor transaction activity at designated points within the processing logic and select transactions for subsequent audit review. The auditor provides specifications and requirements to the application development team during the design of the application so that the SCARF control points can be inserted in the processing logic where errors or failures are most likely to occur. The SCARF computer program code reviews each transaction being processed and selects those that meet criteria established by the auditors. The selected transactions are recorded on a special log, which is then processed to produce a SCARF report for the audit group.

Figure 16-2 is an output report produced from a SCARF log. The report contains the following information:

- 1 Date the report was printed (*November 27, 198X*)
- 2 Report identification number assigned to this particular SCARF application (*ASRO90*)
- 3 Source of the data that appear in this report (file *INFO82*)
- 4 The complete record image (twenty characters) of each transaction that was recorded in the SCARF log
- 5 A message that explains the reason each transaction was selected for recording in the SCARF log
- 6 The date (*November 26, 198X*) on which each transaction was written in the SCARF log
- 7 The time (twenty-four-hour clock expressed in hours, minutes, and seconds) at which each transaction was written in the SCARF log

SCARF REPORT - 11/27/8X REPORT NUMBER: ASR090 RECORD FILE: INFO82

<u>SCARF INPUT RECORD</u>	<u>MESSAGE</u>	<u>DATE</u>	<u>TIME</u>
7349300002811268X090	INVALID RECORD	11268X	143116
9438A00100011268X090	INVALID RECORD	11268X	144343
4329 00300011268X090	INVALID RECORD	11268X	144859

Figure 16-2 SCARF output report

## 258 Computer Audit Techniques Used Concurrently With Processing

As an example of a SCARF application, an accounts receivable system requires every transaction accepted for processing to contain the account number of a valid account on the master file. In certain circumstances it is desirable or necessary to override this control, which may be accomplished by a supervisory approval in the form of a high-level password. In this case, the auditor is interested in reviewing any transaction that exceeds \$5,000 and bypasses the account validation control. To meet this audit objective, additional SCARF coding is added to the program logic at each point at which a control override is recognized. The SCARF code identifies any transaction that involves an override and exceeds the \$5,000 limit and writes it in the SCARF log.

When using the SCARF technique, the auditor must establish the point at which the SCARF review will be employed. Factors to be considered include the following:

- The accounting policies and procedures used by the organization. SCARF coding might be inserted whenever program logic makes critical decisions regarding compliance with management policies and procedures for the recording of transactions.
- Industry practices for handling unusual transactions. In such cases, it might be useful to insert a SCARF review code at those points within the program where special handling is provided for certain types of transactions.
- Common controls implemented by the EDP or operations groups. For example, normal processing for an application might be designed to recognize five different types of transactions. The program logic might also provide a routine for handling transactions that do not contain the expected code values. Since these exceptions represent a variety of error conditions, ranging from input preparation errors to deliberate attempts to manipulate the system, auditors might want to insert a SCARF review point within the program to log any significant transaction that does not contain a valid code value.
- Security features within the system. For example, a SCARF review point might be inserted in the password verification routine. In this way, any unusual password activity, possibly indicative of an attempt to subvert the system, would be recorded on the SCARF log.
- Heavy input/output or processing activity. SCARF points may be used to identify transaction data that place an unusual demand on system resources. This might serve as the basis for developing some recommendations for improving the efficiency of EDP operations.

Other types of activity that might be subject to SCARF analysis include input errors, out-of-balance conditions, detected errors that are overlooked by the system (to facilitate processing), corrections generated by the system, or unusually large financial transactions.

Before implementing SCARF in a particular computer application, the auditor should obtain management approval. Since implementation requires the use of systems design and programming resources, the auditor should obtain appropriate authorization for the allocation of resources. The auditor must become involved in the actual design of the SCARF implementation, which involves selecting the applications to be implemented, developing specifications for the review points, and specifying the data to be recorded on the SCARF log.

Dynamic logging with SCARF can be a very powerful tool for monitoring unusual activity within a system, and it has proven to be quite successful. Some users of the approach feel there is a tendency on the part of auditors to set too low a threshold on the kind of activity to be written on the log, which results in the generation of a high

volume of SCARF records. SCARF is intended to highlight unusual transactions and subject them to audit review; therefore, effective implementation of the approach requires a realistic appraisal of thresholds. If, after the system is operational, too many exceptions are recorded, the auditors should consider revising the parameters used to generate entries on the SCARF log.

### ON-LINE AUDIT MONITOR

In many on-line systems, input flows through a standard set of validation and update modules or programs. Validation performed by this set of modules is controlled by a centralized validation facility. To support this approach, a validation instruction table, file, or database is created, which contains descriptions of the criteria to be used in performing validations. Each possible transaction or transaction type is defined in terms of the specific validation criteria to be applied.

The validation instruction database or file can be designed with extra fields to be used by the auditor in specifying additional criteria to be applied. This approach has been termed an on-line audit monitor. For example, an account number is normally subject to validation to ensure that the field contains the proper number of numeric characters. As an option, the auditor might specify additional criteria indicating that specific accounts or a range of account numbers are to be subject to audit surveillance. This option is implemented by adding the auditor's criteria to the validation table. In one implementation of this approach, the auditors used their own terminal to enter additional criteria into the validation instruction database.

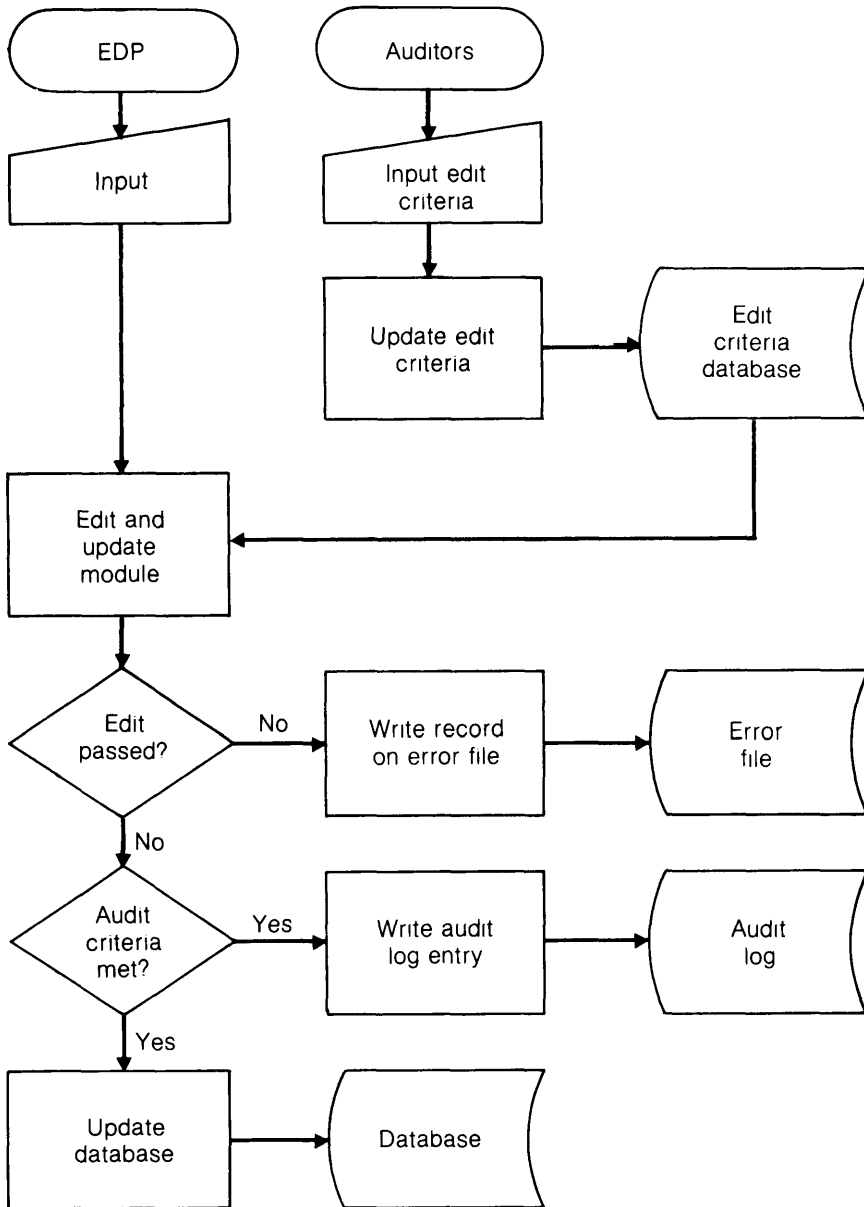
On-line audit monitoring might be performed in the following sequence:

- As a transaction enters the system, the standard validation routines are applied.
- When normal validation has been completed, audit criteria (if specified) will also be applied to the transaction.
- If the transaction meets the criteria specified by the auditors, it is written on an audit log on disk or tape.
- The transaction is marked with a special character or code that identifies it as subject to audit review.
- The transaction is released for normal processing.

The contents of the audit log are normally printed on a daily basis and sent to the auditors so that they have a list of all input transactions that met the criteria they established. If the transactions are marked as subject to audit review, any subsequent activity involving those transactions can be recorded on an audit log. Such activity would include any modification of the transaction or its deletion from the active files. A flow-chart of an audit monitor application is provided in figure 16-3.

In one case the auditors were not able to enter their criteria on-line, but they were able to specify their own validation criteria to be applied to specific transaction types. A transaction that met the auditors' specifications was immediately printed out on an on-line terminal located in the auditors' office. This allowed the auditors to perform an on-line surveillance of unusual or exceptional transactions entering the system.

The on-line audit monitor approach gives auditors a powerful tool for the selective review of system activity, however, it should be applied selectively, using criteria that do not generate high volumes of audit output. If too many transactions meet auditor specifications, the additional processing could have a serious impact on system performance.



**Figure 16-3** On-line audit monitor

## Transaction Selection and Process Data Recording

The transaction selection techniques described in the previous section print the selected transactions without providing additional data on the processing performed on the transaction. A variation of the simple selection techniques entails the additional recording of process data. Three variations are an audit indicator (AI), an extended record, and a snapshot.

### AUDIT INDICATORS

An audit code or indicator (AI) can be added to preselected input transactions, indicating that the transactions are to receive special handling for audit purposes. As they pass through selected points in the application system logic, the transactions and related process data are written in an audit indicator log. The log provides a trace of the transactions' progress through the application processing system.

In using an AI, the auditor decides how individual transactions are to be marked. The auditor can select transactions manually before EDP entry, in which case the proper code to activate AI processing is then entered as part of the original input for the selected transactions. Alternatively, the audit indicator flag can be automatically added to transactions on a random basis as they enter the processing cycle. The selection of an approach depends on the audit objective in a particular situation.

The auditor specifies the information to be recorded in the audit indicator log. Normally, this includes a copy of the transaction, the name of the computer program that was processing the record when the audit indicator log entry was generated, and the date and time of the entry. The log itself can be a special file written for audit purposes, a part of the error file, or a part of the system accounting file.

Figure 16-4 illustrates an output report produced from data on an AI log. The report contains the following information:

- 1 Date the report was printed (*November 27, 198X*)
- 2 The complete record image (twenty characters) of each transaction that was recorded in the AI log
- 3 The application program (*INPO82*) that was processing when the transaction was recorded in the AI log
- 4 The AI point within the program (point *0001* in program *INPO82*) that caused each transaction to be selected for recording in the AI log
- 5 The date (*November 26, 198X*) on which each transaction was written in the AI log
- 6 The time (twenty-four-hour clock expressed in hours, minutes, and seconds) at which each transaction was written in the AI log
- 7 The code within each transaction (in this case, anything other than blanks in positions eighteen to twenty in each record) that triggered AI processing for that transaction

In its basic form, the audit indicator technique can be used to log selected transactions in and out of each major program in the processing cycle. It may also be used to record the details of file updates triggered by the selected transactions. All of these possibilities must be considered during initial system design.

The AI approach has several positive features. It facilitates a thorough testing of the system using a small volume of data. The random selection feature allows tests to take place at all times throughout the processing cycle, and it provides a good review of the

AUDIT INDICATOR REPORT - 11/27/8X				
<u>AUDIT INDICATOR RECORD</u>	<u>PROGRAM</u>	<u>LOCATION</u>	<u>DATE</u>	<u>TIME</u>
2392100002011268X239	INP082	0001	11268X	143123
1063200100011268X240	INP082	0001	11268X	144337
3948200999911268X241	INP082	0001	11268X	144353

**Figure 16-4** Audit indicator output report

types of transactions being processed. The technique may also prove useful to systems personnel, if they are encountering a problem with a particular type of transaction, the transactions can be flagged to obtain a record of the related processing. There is a potential danger, however, that AI features may be used to perform unauthorized manipulations of the system, since the method flags certain transactions and processes them in part outside the normal system.

### **EXTENDED RECORDS**

Rather than being written in a log, the audit data items can be added to the transaction record, which is extended for this purpose. Auditors can extract the transactions that have such extended data and prepare reports for their review. While some applications can use this approach, an application designed around fixed-length records will incur a considerable amount of additional overhead to provide the space that will be occupied by the audit data if the record is selected.

### **SNAPSHOTS**

The snapshot technique is more complex than the SCARF or audit indicator approach. Input transactions are tagged, and the path of a tagged transaction through the system is recorded at selected points within the program logic. When an application program encounters a tagged transaction, the transaction is written in an audit log. Along with the transaction data, the snapshot routine also records the date and time at which the snapshot occurred, the identification of the application program involved, an indication of the point in the program at which the snapshot was generated, and any additional data required for the snapshot record. Normally, additional data include the contents of registers, accumulators, or other selected areas of memory related to the processing of the transaction.

In an application program that accumulates totals or performs some type of computation, the snapshot record might include the intermediate results obtained while making the computation or the contents of the accumulators before and after transac-



tion processing. This additional information makes the snapshot approach more detailed than other techniques. All the information included on the snapshot record is written in a log. On a scheduled basis, the contents of the log are printed and turned over to the auditors.

Figure 16-5 is a portion of an output report generated from a snapshot log.

1. Date the report was printed (*November 27, 198X*)
2. Report identification number assigned to this particular snapshot application (*130025*)
3. The application program (*INPO82*) that was processed when the transactions on this report were recorded in the snapshot log
4. The snapshot point within the program (point *0001* in program *INPO82*) that caused each transaction to be selected for recording in the snapshot log
5. Contents of four areas in main storage (*A4, A9, A14, A19*) that were used to accumulate control totals during processing. Related storage areas to be included in the snapshot log must be specified when the audit application is being implemented
6. The date (*November 26, 198X*) on which each transaction was written in the snapshot log
7. The time (twenty-four-hour clock expressed in hours, minutes, and seconds) at which each transaction was written in the snapshot log

Snapshots allow selected transactions to be traced through the application programs. Since the tagging of the input can be under the direct control of the auditor, the tagging and tracing will take place only on those transactions specifically selected by the audit group. It is possible to incorporate the transaction tagging as part of normal input processing. Tags can be assigned in accord with established criteria or on a random basis.

In some cases, a snapshot has proven to be a powerful systems analysis tool. The ability to monitor selected transactions and simultaneously record the contents of specific areas within memory can benefit the systems group when they are debugging program logic.

The snapshot technique is implemented as part of the original design of the system. The auditor becomes involved in the design and is treated as a user of the system. He specifies requirements and then works with the systems design and implementation groups to see that those requirements are met.

## **Audit Processing Facilities Within Applications**

In the design of applications, audit requirements can be included in the system requirements. Some audit requirements can be satisfied by special audit routines that are part of the application programs but are coded as separate program modules. Alternatively, the auditor may specify future needs that are not coded immediately; instead, the application is designed to allow them to be added.

### **AUDIT MODULES**

In applications involving high volumes of accounts or transactions, it is often effective to place audit software modules within existing client application programs. For example,

SNAPSHOT OUTPUT REPORT

POINT	SNAPSHOT TRANSACTION	SNAPSHOT ID:	PROGRAM ID:	INPD82	DATE	TIME
0001	7349300002811268X090	130025			11268X	143116
	A4-REJECT-NO-TYPE	0				
	A9-UPDATE-NO-TYPE	0				
	A14-COMBINED-NO-TYPE	0				
	A19-OVERALL-NO-TYPE	28				
0001	943800100011268X090				11268X	144343
	A4-REJECT-NO-TYPE	28				
	A9-UPDATE-NO-TYPE	0				
	A14-COMBINED-NO-TYPE	0				
	A19-OVERALL-NO-TYPE	1,028				
0001	4329 00300011268X090				11268X	144859
	A4-REJECT-NO-TYPE	1,028				
	A9-UPDATE-NO-TYPE	0				
	A14-COMBINED-NO-TYPE	0				
	A19-OVERALL-NO-TYPE	4,028				

Figure 16-5 Snapshot output report

the client's monthly billing program might include an audit routine to select accounts randomly for confirmation and print the confirmation wording on the monthly statement. This audit module remains in the client program and is executed during normal processing. By the use of appropriate commands, the auditor activates the module and performs audit processing as part of the normal client run. The additional coding within the application program has little effect on the client's normal processing, but it allows the auditor to perform additional work whenever it is required.

A disadvantage of the audit module is that the logic of the auditor's processing is incorporated within a client program and is therefore available for examination by client personnel. This may make it easier for someone within the client's organization to subvert the auditor's processing. While this is a theoretical disadvantage, in most cases it is not a serious threat, since the kinds of processing that would call for an embedded audit module are reasonably straightforward and do not contain highly sensitive information.

### **AUDIT HOOKS**

Audit hooks are points in application programs, called "exits," that allow an auditor to insert commands for special audit processing. This allows an auditor to modify a standard application or systems program to perform processing needed to support an audit activity. For example, a utility program used to reorganize a database might provide an audit hook for the auditor to insert additional coding to accumulate totals and record counts from the database. This allows the auditor to obtain control totals independently as a by-product of normal system processing.

The use of audit hooks is particularly valuable in dealing with application software packages. Since there is a trend away from individually written software to the use of packages, the audit hook approach may become very significant in the future. Ultimately, the availability of audit hooks in package software will depend on the need for them, so that installations will be willing to pay the additional cost for providing and maintaining audit exits within application and system software.

### **Summary**

This chapter describes eight techniques for selection and testing for audit purposes while normal processing is being performed. These concurrent techniques fall into four process classes: testing of programs and procedures, transaction selection, transaction selection with recording of process data, and audit processing facilities in applications.

The concurrent techniques for selection and testing are more likely to be used by internal auditors, but independent auditors should be familiar with them in order to evaluate the work of internal auditors and to select a technique if it is appropriate. Although the techniques have received relatively little use, they can be very effective.



---

# Audit Tools and Techniques for Database Management Systems

# 17

A database management system (DBMS) is a software system used to manage the storage, updating, and retrieval of data from one or more databases. The control characteristics of a database management system are described in chapter 7, this chapter describes audit procedures applied to significant financial records maintained with the use of a database management system. As the term is used in this chapter, the database includes the use of shared data and data independence from applications; it excludes database management system software used for managing single application files. If a database management system is used to manage access in an application that has its own files, the procedures discussed in this chapter may not be applicable.

The audit procedures and techniques employed for a database management system can be conveniently classified as follows.

- Study and evaluation of internal accounting control for the database and database management system
- Compliance testing of database controls
- Substantive tests of data maintained by the database management system

## The Auditor's Study and Evaluation of Internal Accounting Control

Before beginning an audit, the auditor must gain a general understanding of the controls over the database system and the database management system. Three important sources of information for understanding the database system are—

1. The database administrator, who is responsible for designing, maintaining, and monitoring the database system and for establishing controls over backup, access, and use of the data, and for maintaining the software
2. The data dictionary/directory, a software facility that organizes and keeps track of the list of data items in the database. The data dictionary can provide the auditor

## 268 Audit Tools and Techniques for Database Management Systems

with up-to-date documentation of the database, including the following for each data element

- The definition
- The programs that use the data element
- The edit and validation criteria
- The security and access limitations

- 3 Database system documentation, including the vendor's literature Because the organization may not implement all the control features listed in the vendor's documentation, the auditor should determine the control features available in the DBMS and the features that have been implemented in the system under audit

As part of the review of internal accounting controls for the database and DBMS, the auditor may wish to evaluate the following controls

- Organization and operations, including the segregation of functions for the database administrator and the assignments of responsibility for data
- Documentation standards for data definitions and database management system software
- Controls over access to and changes in the programs and data
- Other database controls of data validity, such as edit and validation procedures Since standardized modules can perform most of the edit and validation procedures, the auditor may not have to review those procedures separately for each application being reviewed The review and testing of the DBMS validation module can provide a basis for reliance on the edit and validation control whenever it is used

### **The Use of Database Management System Facilities in the Review of the System and in Testing**

In auditing in a database environment, the auditor will need to plan well in advance to make sure that the necessary data is available by asking the database administrator to retain the records Such a request may simply require retaining copies of the systems activities logs, or it may involve retaining a copy of a complete segment of the database as of each accounting cutoff date and date of interim testing

The volume of data to be made available through the database system is another important consideration The auditor may access the data through the database management system, or he may have some or all of the database copied for audit testing In a very large database, however, merely copying the database onto a storage medium, such as magnetic tape, may take several hours If part or all of a database is to be copied for audit, arrangements must be made with the database administrator to ensure that the information will be in a form that is efficient and effective for the auditor's use In addition to the data sequence and format, the storage media also should be considered For example, magnetic tape might be easier to use if the auditor decides to remove the retained data from the installation in order to establish audit control or to perform audit processing at another installation

The database management system has several facilities that can be useful audit tools

- Data dictionary/directory
- Backup and recovery facilities
- Testing and debugging aids
- Query facilities

### **DATA DICTIONARY/DIRECTORY**

Chapter 7 discusses the functions of the data dictionary. The dictionary can be helpful to the auditor in several ways:

- Helping the auditor to gain an understanding of the database system. The data dictionary maintains a record of data definitions, the relationship of each data element to application programs, and the security and validity controls over the data items.
- Assisting the auditor in testing controls within the application programs. Many data dictionary/directory packages can generate test data in the format required by each application program.
- Providing an audit trail. The data dictionary can log changes to data descriptions, prepare reports showing which applications used each data element, and provide other database usage reports.

### **BACKUP AND RECOVERY FACILITIES**

The backup and recovery facilities for a database consist of routines in the DBMS to perform the following tasks:

- Dump a copy of the database on a backup storage medium.
- Log all changes to the database.
- Process the backup copy with the log of changes to recreate the database.
- Restore the database to the state in which it existed at some point in the past by eliminating each of the updates that occurred after that point.

The logs are based on the before-and-after image principle. For every update, two records are written in the log. One is the image of the record as it appeared before the update, and the other is the image after the update.

The database log can be a valuable audit trail. It may be useful for tracing transactions through the system and for tracing the source of a change in a data item. The database recovery log is likely to be essential in any attempt to determine the updating activity that has taken place.

However, there are often practical difficulties in the use of the log for audit purposes. These involve the volume and format of log records.

For very large databases, the volume of log records can be significant, perhaps several reels of tape per day. This volume does not preclude use for audit analyses, but it does suggest that the auditor must be selective and have clear objectives for the analysis.

Some auditors have encountered difficulties in using the log because of the format used to write them. These formats were not designed to interface with computer audit software or other common data retrieval tools used by auditors. However, there are

## 270 Audit Tools and Techniques for Database Management Systems

database utility programs that are specifically designed to interface with the recovery log in order to extract records and prepare reports of recorded activity. An auditor working in a database environment can become familiar with the utility programs to access the recovery log files and use them to prepare reports on the processing taking place within the system and reports of changes in specific data items.

Another use of the database backup and recovery facilities is to obtain a copy of the database for testing purposes. For example, if the auditor wishes to use test data to examine database processing procedures, he must be careful not to make changes to the database that may inadvertently go undetected or that may be difficult to reverse. To prevent inadvertent changes to the database, the dump/restore facilities can be used to create a duplicate copy of the relevant part of the database, and the auditor can process transactions to test the controls within the application programs that use the database. After testing is completed, the modified database is removed, and the original database is restored.

This audit technique is also useful for testing the company's database recovery procedures as part of a review of continuation provisions. These procedures are not a direct audit concern; however, they may be included in the audit because the ability to recover the database in case of an error or system failure is vital when company financial records are stored in the database. By performing the copy-and-restore procedures under carefully controlled conditions, the auditor can test whether the recovery procedures are in operation and whether they restore the database accurately and completely. If the procedures are weak, the auditor may wish to suggest improvements in a management letter.

### TESTING AND DEBUGGING AIDS

The facilities designed to help programmers and the database administrator set up the database system, test the system, and eliminate errors can also be useful for audit tests. Three types of facilities are of interest to the auditor:

#### Facilities to Test Linkages and Pointers

Database management systems use internal pointers and linkages to associate related data items. Software facilities to check that the linkages are correct are available in many database management systems. For example, the facilities may check that pointers reference existing records or that pointers for a particular data element reference only record types related to that data element. In addition, software facilities may check the linkages by verifying the forward and backward chaining process to ensure that the pointers in a forward chaining technique reference the same records as those in a backward chaining technique (figure 17-1).

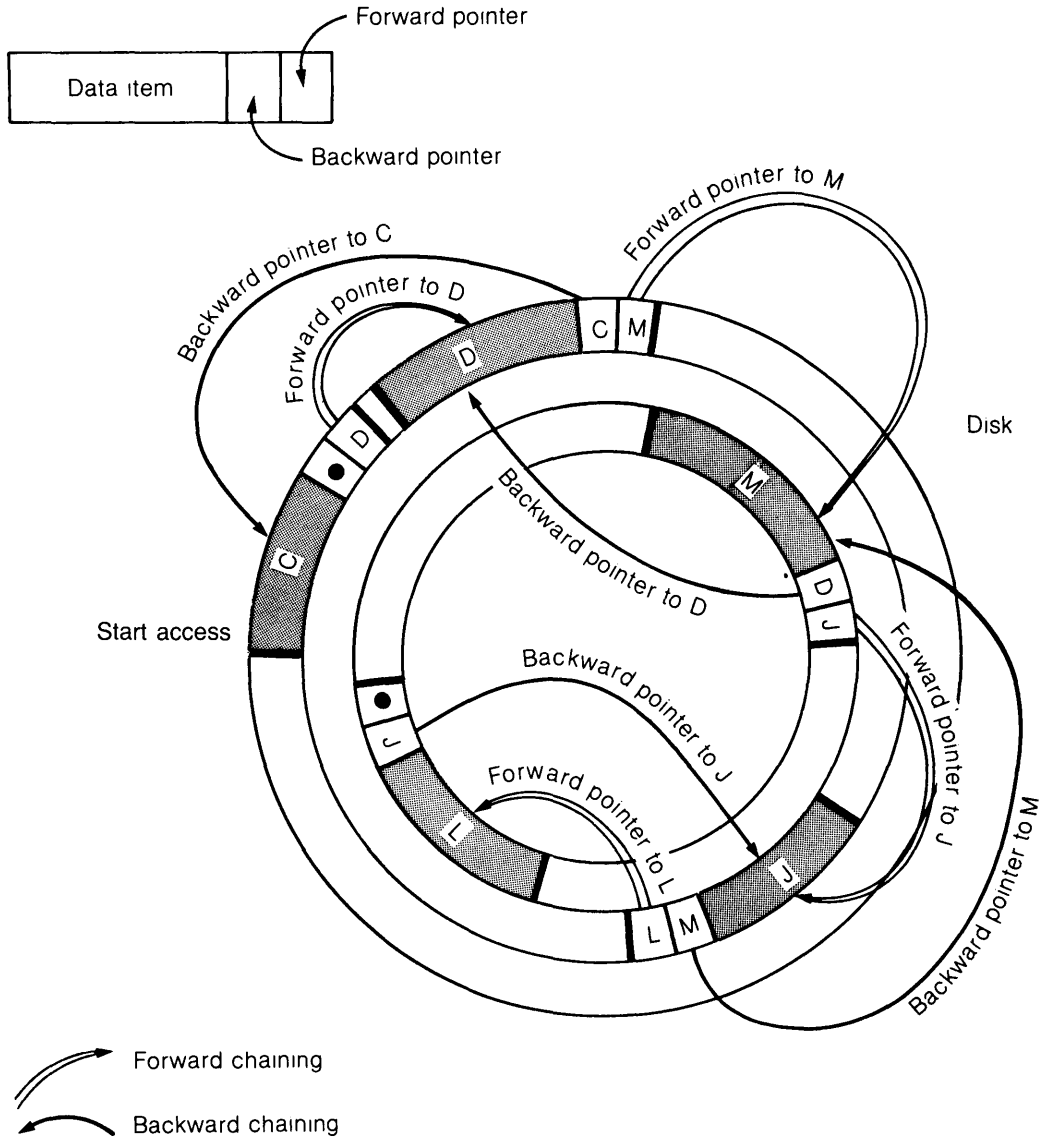
The software facilities to test linkages and pointers generate a report describing any errors, identifying the data item in error, and identifying the name of its owner. Thus, the auditor can review the latest linkage and pointer report and trace errors to their causes and to their resolution. If the reports are not current, the auditor may decide to generate updated linkage and chaining reports.

One audit objective may be to determine that the financial records stored on the database are complete. The auditor can test that the accessed data supports a particular balance, however, it is difficult to verify that the balance reflects all the appropriate



The data to be accessed is —

- First record (start)            C
- Second record                 D
- Third record                    M
- Fourth record                  J
- Fifth record                    L



**Figure 17-1** Forward and backward chaining

## 272 Audit Tools and Techniques for Database Management Systems

data The database testing facilities can provide some assurance that the data records used in an audit test are complete and that all items in the test have been accessed

### **Trace Facilities**

When the database management system trace facility is activated, it records the step-by-step activities of the database The database trace facility may be applied as an audit technique concurrent with processing to provide audit data on database processing

### **Operation of the Database Management System in Test Mode**

Some databases allow operation in a test mode, in which the activities and results of processing are temporarily recorded and reported but no changes are actually made to the database These facilities can aid the auditor in the use of test data for compliance tests, they are an alternative to the test database produced by backup and recovery facilities

### **QUERY FACILITIES**

A serious consideration for the auditor working in a database environment is how to access the information stored in the database Auditors often use generalized audit software to access data stored in an EDP system, without relying on the company's software, however, most generalized audit software cannot access database structures directly There are several solutions to this problem, one of which is for the auditor to use the database query facilities to perform many audit software functions In deciding whether to use the query facilities, the auditor should consider (a) the extent to which the query facilities provide the capabilities to perform the audit test and (b) the reliability of the client's query software

### **Query Audit Capabilities**

Query facilities in database management systems vary in the quantity and variety of their capabilities Some perform almost all the functions of an audit software package, including stratification, record selection based on criteria furnished by the auditor, and mathematical computations Others have very limited capabilities but can be used by the auditor to extract a data file from the database The auditor can then use generalized audit software on the extracted file to perform the necessary tests

### **Audit Reliance**

The auditor should consider whether the query facilities can be relied upon for record selection and access purposes In deciding whether to use the client query facilities, the auditor should consider whether information obtained through the query facility can be reconciled with another source and whether the risk of relying on the query facility is high enough to justify a detailed review of the query facility performance In most cases, a detailed review of the query facility is not necessary because query software is supplied by an outside vendor Few employees have access to the program source code or have detailed knowledge of its coding, which reduces the risk of tampering

## **The Use of Generalized or Customized Audit Software With a Database Management System**

The techniques of generalized and specialized audit software can be helpful in a database environment, however, the difficulties caused by the complex data storage structures require some modifications in those techniques. Three approaches to access and audit analysis are (1) to copy the relevant portion of the database and process this file using audit software, (2) to modify generalized audit software to access data through the database management system, and (3) to use specialized audit software.

### **COPYING RELEVANT RECORDS ONTO A SEQUENTIAL FILE**

One method for analyzing financial records stored on a database is to request that all relevant records be copied onto a sequential file that can be accessed and analyzed with standard generalized audit software. This method relies on the installation personnel and the installation DBMS procedures for transferring the records; the auditor obtains no evidence regarding the records as they exist in the database. If this technique is used, there should normally be a general ledger control or similar control figures against which to test the completeness of the audit file copied from the database.

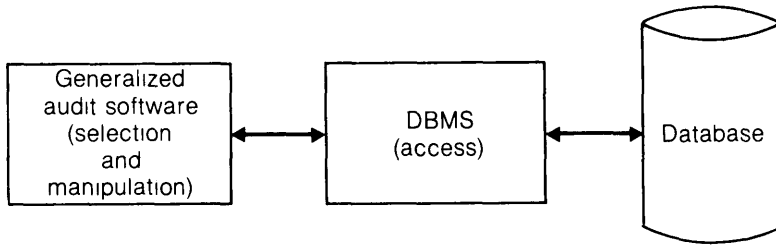
### **MODIFIED GENERALIZED AUDIT SOFTWARE**

Some generalized audit software systems can interface with a few of the more common database management systems (figure 17-2). In this approach, the audit software program accesses the database like any other application program. Its data requirements are defined and described to the DBMS. The audit software program issues a request to the DBMS for the record it needs, and the DBMS accesses the data record and provides it to the audit program. This method has the advantage of relying on the client's query software for access to the data but not for the selection and manipulation of the data. As an alternative, if the generalized audit software cannot interface with the DBMS, the auditor can write a computer program that uses the DBMS to access the database, which operates the same as any other application program. It has many of the same advantages and disadvantages of generalized audit software.

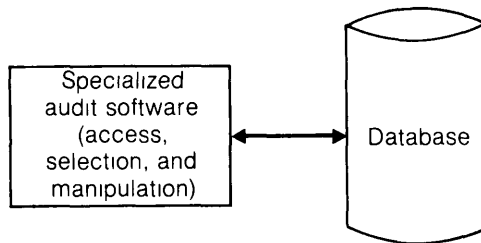
In order to use generalized audit software with the DBMS, the auditor should (a) notify the database administrator early in the audit to arrange for the testing and use of the generalized audit software package and its interface and (b) compare the descriptions of the data that the audit software will request with the application system data description and data dictionary to make sure they match.

### **CUSTOMIZED AUDIT PROGRAMS**

To access the database directly, the auditor may write a computer program to perform the required audit tasks. Customized programs have the following three implications in a database environment:



Accessing the database through the database management system



Accessing the database directly

**Figure 17-2** Approaches to using generalized audit software in a database environment

- 1 The complexity of the data structures may make the writing of the program difficult and time consuming. Therefore, substantial technical expertise is required.
- 2 For control reasons, the client may prohibit any access to the database except through the DBMS.
- 3 The computer program should contain extensive controls to prevent inadvertent update or damage to the database.

The risks of customized software to access the database directly may exceed its benefits unless there are indications of database problems that cannot be investigated in any other manner.

## Selecting Appropriate Audit Procedures

As a preliminary step in his review, the auditor should study the organizational controls in the system. The auditor should know the responsibilities assigned to the database administrator, the responsibility of financial accounting for financial records, and the available controls over unauthorized modifications of DBMS software.

The auditor should become familiar with the database dictionary or data dictionary facilities of the DBMS and should examine data descriptions for the data items relevant to the financial records. He should become familiar with the facilities and controls for data validation, updating, and query, and he should perform test queries on financial data items relevant to the audit.

If the organizational controls appear appropriate, the auditor should then decide on the compliance tests to be performed on use of the DBMS. The following are some examples of the tests to be performed:

- Examination of documentation that demonstrates controls in operation
- Tracing of input and updating procedures through documentation of processing
- Use of query facilities to verify that sample updating transactions are processed correctly and completely

If compliance tests need to be more extensive, other techniques may be used, such as log analysis, linkage and pointer review, a test run with trace facilities, testing of data in a test mode, or testing of data with a test database. These techniques require some database expertise, and they should be performed in connection with well formulated objectives for their use.

The auditor must also decide on substantive tests of financial data maintained by the DBMS. He should select the method for obtaining records from the database and for performing analyses. If the database appears to be well controlled, the auditor may decide to use the DBMS query facilities (alone or as a method to obtain files for processing with audit software), modified software to access data through the DBMS, or a special audit program to interface with the DBMS. On the other hand, the auditor may use a direct access method if these methods are not feasible or if special testing is desirable and personnel with the necessary expertise are available.

Finally, the auditor may review the backup and recovery facilities and the evidence of effective operation.

## Summary

When auditing in a database environment, the auditor has three important additional sources of information: the database administrator, the data dictionary/directory, and the database systems documentation.

The audit techniques discussed in previous chapters are valuable in a database environment; in addition, the database management system may provide facilities that can be used as audit tools. These facilities include the data dictionary/directory, backup and recovery facilities, testing and debugging aids, and query facilities.

Generalized audit software and generalized audit programs can be used in some situations. Relevant financial records may be copied onto a sequential file to be analyzed by audit software, generalized audit software may be modified to access data through the database management system, or specialized audit software may be written to access the financial records on the database, either directly or through the DBMS.



---

SECTION V

# Other Data Processing Environments and EDP Auditing Activities

The preceding sections have emphasized the internal controls and audit techniques for financial audits when the client organization has in-house computer data processing facilities. This section discusses the use of outside data processing services, the use of small computers, and staffing for EDP audit activities.

Chapter 18—Auditing Service-Center-Produced Records

Special considerations and audit procedures when the client organization uses a service center to perform processing of financial records.

Chapter 19—Control and Audit Impacts of Minicomputer and Microcomputer Systems

Describes the special characteristics of processing and internal control when a minicomputer or microcomputer is used for processing. The chapter also explains audit considerations in this small computer environment.

Chapter 20—Audit Staff Selection, Organization, and Training for EDP Auditing

Describes the problems and alternative solutions for providing technical assistance and review capabilities when performing EDP audit activities.





---

## **Auditing Service-Center-Produced Records**

# 18

Computer service organizations provide a variety of services related to the processing of financial records. They are used by organizations that do not have enough data processing activity to justify having a computer or do not want to invest the time and resources required to establish and operate a computer-based data processing facility. Companies with their own computer facilities may use service organizations to provide additional capacity to handle peak loads, to supply a backup capability in the event of hardware failure or natural disaster, or to perform special services that might not be suitable for the in-house computer. A computer service company may supply software packages or contract programming, assist in the maintenance of normal processing activities during a conversion from one computer to another, enter into a contractual arrangement to provide a standby facility for a limited number of clients, or serve as a broker for the sale of excess computer time or capacity. Some computer service organizations provide facilities management activities, in which they accept full responsibility for the management of another organization's in-house computer installation. These services may be provided by an organization formed exclusively for that purpose, by a division of an equipment manufacturer, by a separate subsidiary of a bank or other financial institution, by the data processing subsidiary of a large corporation or conglomerate, by a CPA firm, or by other service companies.

Some service centers also sell time on their equipment, so that users may operate their own programs with their own personnel. From an auditing and control viewpoint, it makes no difference whether the equipment belongs to the client or to a service center as long as the client organization operates its own programs with its own personnel. The audit and control problems and programs discussed in this chapter, therefore, are applicable only when the center provides processing services, not when the center merely rents a block of time on the EDP equipment.

Time-shared data processing, in which many separate users are connected directly to a central computer, is another form of computer service with some unique characteristics. It is discussed in a separate section of this chapter.

## Operations of a Computer Data Processing Service Center

In the most common type of service involving financial records, the service center processes various types of accounting transactions. Three types of application programs are used for this processing:

- 1 Standard programs developed by the service center for specific functions, for example, payroll preparation
- 2 Standard industry programs, such as customer and stock record accounting for the brokerage industry
- 3 Programs tailored to the client's specific needs that have been prepared by the user, by the service center organization, or by another organization

In a typical service, the user submits the input transaction data to the computer center for processing. The input data may be in hard copy documents or a machine-readable medium, such as a diskette or cassette prepared by the user. It is usually delivered to the service center by messenger or by mail, though in some cases input data items are sent via a telecommunications system. The amount of input can sometimes be reduced by the use of the exception principle. For example, in a salaried payroll operation in which most employees are paid the same amount each period, only the changes (exceptions) from the previous payroll need be reported for the payroll to be processed.

The permanent (master) files against which the input is processed are usually retained by the computer service center. A master file may be retained on tape, disk, or other storage medium, depending on the system and equipment being used. Changes in the master file data are usually submitted as file maintenance adjustments and processed either before or concurrently with the processing of the transactions.

Most computer service center processing is designed to provide the user with the final output document (invoice, payroll check, and so on) together with the appropriate control totals and listings or registers. Occasionally the end product of the processing consists of machine-readable records that must be processed further or printed at the user's data center. Service centers generally produce reports according to an established schedule. For an added fee, service centers will provide special-purpose reports, though these require some time because of scheduling problems or the need to write a new report program. The data stored in the master files are not commonly reproduced for distribution to the user, except in response to a special request.

The responsibility for source records and the retention of output records usually rests with the user. The service center retains only those records necessary for the reconstruction of files in the event of loss, destruction, or significant error. Therefore, the report obtained from the service center must contain all necessary historical and identification data, and the user must arrange for proper retention and storage of source documents and reports. Users should also be concerned about compliance with IRS Revenue Ruling 71-20 on record retention. (See chapter 5.)

## Responsibilities and Controls at the Service Center

The computer service center normally is responsible for the software employed in the data processing services that it provides. If it provides a client with standardized applications developed by the service center for a number of users, the client is responsible

for understanding the application well enough to provide necessary input, respond to reported exceptions and differences, and understand the output records. If specialized applications are developed for a single client, the service center participates (and often takes the initiative) in the design and accepts full responsibility for programming, program testing, and operation, the client is responsible for defining the requirements of the application and specifying user-oriented features of the design. In addition, user involvement should include review and approval of the system at key points within the design cycle.

The computer service center usually assumes full responsibility for hiring, training, and supervising the personnel who program and operate the system. The user seldom has any voice in these matters, in fact, the user is often not informed about the qualifications of the personnel at the service center and has contact only with an account executive or other liaison personnel.

Users of a computer service center are often concerned about the security measures taken to protect the information held at the center. A user organization may be able to satisfy itself about the adequacy of the basic security controls through inquiry and observation. If security and confidentiality are significant concerns, however, the user can employ additional measures to reduce exposure. For example, users may furnish the service center with code numbers instead of names for customers or employees, users themselves retain responsibility for inserting the names on the output. Users can deliver the input data, observe the processing, and, at the completion of the processing, withdraw all transaction and master file data from the center.

Another important aspect of security involves the provisions at the computer service center for master file reconstruction in the event of loss or destruction. Methods used for in-house installations, such as fireproof vaults, off-premises storage, and so on, apply equally to service centers. The user should be concerned about these provisions and about the arrangements the center has made for backup facilities. The question of backup may be critical if the service center obtains its computer time by off-shift rental of computers belonging to other organizations.

The audit trail for service center processing should not be a problem, since the listings and other output provided by the service center should provide adequate documentation of data processing activity. The printouts and other data retained by the user should be sufficient for the reconstruction of processing in the event of a computer center failure. If a service center provides on-line processing, there should be backup and recovery provisions for both service center and user activities in case of a failure in the data communication facilities.

The agreement with a service center should specify ownership of files and programs. Client files maintained by the service center should belong to the client. Ownership of programs developed for the client by the service center should be clearly specified. The service center may wish to use the program with other clients, and this option should be negotiated. If the client has made full payment for a special application, the service center should deliver complete documentation, including source code listing.

## User Controls

The controls that ensure the orderly and supervised processing of data are the responsibility of the management of the computer service center. The user is responsible for establishing controls over data transmitted to the service center, establishing procedures for checking the quality and completeness of processing, providing for recon-

## 282 Auditing Service-Center-Produced Records

struction of source documents in the event of loss in handling, and establishing controls to ensure timely correction.

Users generally establish some overall input data controls (control totals, document counts, numbers of accounts, and so on) to enable them to check the completeness and accuracy of service center processing. They keep a user log of the control totals for source data transmitted for processing, records in master files, and some data items in master files. The control totals from processing are also recorded in the log, and the differences are investigated and reconciled. Users may review all output documents and records or a sample of these records, depending on the nature and volume of the processing. The users may also undertake to test check manually some of the processing performed by the service center.

The users should make provisions to protect against a loss of source documents in transit to and from the service center. For example, if paid but unposted checks are transported to a service center for bank data processing, a copy should be kept on microfilm at the sending bank. Another approach is for a copy of the source documents to be prepared for data processing purposes.

The user usually reviews the input data items before sending them for processing. Any errors or omissions detected in this review are corrected before the source data are sent to the service center. The center reviews the data items received and subjects them to further input validation routines. Any erroneous data items detected by the service center review or computer input validation routines are listed and returned, unprocessed, to the user.

The user may correct the data items and resubmit them with the next batch or, as in the case of payroll, may have to process the items manually. If processing is performed manually, an adjustment must be prepared to update the master file at the service center before the start of the next processing cycle.

## **Auditor Concerns When a Service Center Is Used for Financial Records**

If a company uses a computer service center, the auditor is faced with the additional concerns of an outside organization's involvement in data processing, internal control, and record retention. However, some of these complicating factors may help improve control, since the outside service center can represent a formalization and, possibly, an expansion of the controls afforded by segregation of functions. Deliberate manipulation of records is made unlikely by the separation of the persons in a position to perform such manipulation from those having custody of or access to the assets of the company.

An exception to the control provided by separation of functions between the user organization and the service center occurs when the service center's employees have access to the user's assets through their status as customers of the user organization. For example, an employee of a service center may have an account at a bank for which the service center performs processing. In such cases, suitable precautions should be taken, which might involve identifying all accounts controlled by service center employees and reviewing all significant transactions within those accounts.

The use of an outside computer service center in operations that can be subjected to overall controls and review, such as payroll preparation or accounts receivable

processing, improves internal control and diminishes audit problems. Some auditors have concluded that, if overall controls can be maintained effectively, neither the client nor the auditor need have great concern about the operations within the service center.

However, if the client's financial records are processed by a service center, the auditor still has a responsibility to review the system to obtain an understanding of such matters as the flow of transactions through the accounting system (SAS 3, par. 25). If the auditor plans to rely on internal accounting control, he should be concerned with user control over data preparation, provisions for completeness and quality assurance, user backup, user error handling procedures, and service center operations

Practical limitations may make it difficult for each auditor of service center customers to satisfy himself directly about the operation and controls within the service center. The client is generally only one of many companies using the service center, so it may be difficult to obtain the cooperation of the service center's personnel. To relieve service centers of the burden of having to deal with auditors for each of the center's clients, one CPA may review the systems, procedures, and controls of the center and provide the results of that review to the auditors of service center clients. This review is termed a *third-party review*

### **Third-Party Reviews of the Service Center**

The AICPA Auditing Standards Board has issued SAS 44, *Special-Purpose Reports on Internal Accounting Control at Service Organizations*, which provides general guidance to user auditors and third-party reviewers (termed *service auditors*). The AICPA has also published an audit and accounting guide entitled *Audits of Service-Center-Produced Records*, which presents the concept of third-party reviews as applied to service centers.

In the third-party review, the auditor studies and tests the service center's descriptions of the system, procedures, and controls. The third-party auditor would include in his report a description of the system used to process customer transactions and the related controls, and a description of the specific control objectives to be achieved. His report describes the scope of the review and the extent of his tests and expresses an opinion about whether the existing systems, procedures, and controls are sufficient to achieve the specific control objectives identified in the description.

The opinion of the third-party auditor can address either "the design of the system" or the design of the system and the proper functioning of that system for a period of time (termed a "report on both the design of the system and compliance tests that are directed to specific objectives of internal accounting control"). The auditor would usually disclaim any opinion on the service center's own accounting controls related to its financial records. However, the auditor would report on any material weaknesses uncovered during the examination.

Even though a third-party review has been performed, a user auditor must exercise professional judgment in deciding whether the report issued by the third-party auditor is appropriate for his purposes and whether the procedures performed by the service auditor included compliance testing of any controls the user auditor intends to rely on. The auditor must also determine the extent of any additional audit procedures that are required. The user auditor is not relieved of responsibility for exercising professional judgment, and the user auditor's report should not refer to the report of the third-party auditor.

## 284 Auditing Service-Center-Produced Records

The third-party review concept has evolved slowly, but third-party audit reports and opinions are being prepared and are proving helpful to user auditors. A survey of CPA firms having experience with third-party audits revealed the following information<sup>1</sup>

- Sixty percent of user auditors made direct contact with the third-party reviewer (or service auditor). Of these, about 50 percent reviewed the third-party review program, and 33 percent reviewed actual working papers. After the review, about 15 percent of the user auditors asked for an extension of procedures.
- Sixty percent of the user auditors felt it was necessary to visit the service center even though there was a third-party reviewer's report.

### Auditor Activities

An auditor whose client uses a computer service center and who plans to rely on internal accounting control must perform three tasks:

- 1 Evaluate user controls over processing.
- 2 Evaluate service center controls over processing, which may involve obtaining and evaluating the report of a third-party reviewer.
- 3 Obtain evidential matter.

### EVALUATION OF USER CONTROLS

The absence of a client's control over the actual processing increases the significance of its controls over input and output. When the client's records are processed at a service center, the auditor generally evaluates the following controls:

- User control over data transmitted for processing, with additional controls necessary if the client performs data conversion
  - Document counts
  - Transaction counts
  - Control totals or hash totals
- User control over master file changes
  - Control printout of all changes
  - Control count of master records
  - Control totals for items in master records
  - Password control over those who are authorized to submit master file changes
- User control over error correction and resubmissions
  - Error printout identifying all errors
  - Log maintained by the client to record the disposition of each error
  - Correction and resubmission procedures
  - On-line editing and data validation
- User control over output
  - Output distribution list of authorized recipients
  - Quality control tests performed on a sample of the output
  - Comparison of control totals on output with input control totals and other control figures, such as ledger account totals

---

<sup>1</sup> Robert S. Roussey, "Third-Party Review of the Computer Service Center," *Journal of Accountancy* (August 1978) 78-82.

- Adequate audit (management) trail for the output
  - Adequate processing references in the transaction output
  - Ledger (master file) records containing adequate updating references
  - Periodic printout of ledger balances
- Adequate protection and security
  - Client retention of a copy of the source documents transmitted to the service center
  - Client provision for file reconstruction

In addition, the auditor may investigate the service center's provisions for routine reconstruction of files and for the security of user records kept at the service center. A sample questionnaire that could be used for recording information about these controls is included as Appendix E to this book.

### **EVALUATION OF SERVICE CENTER CONTROLS**

The user auditor should consider reviewing the controls at a service center when the center is processing financial data that have a material effect on the client's financial statements. The scope of the audit review may not necessarily include the service center itself if the client's controls are considered sufficient to provide reasonable assurance of the reliability and accuracy of the financial data.

If the user auditor has decided to rely upon controls at the service center, he must then decide on the use of the third-party review (if it exists) and the extent of personal observation and testing at the center. Statement on Auditing Standards No. 44 discusses the evaluation of service center controls using the service auditor's report. It explores three issues:

- 1 Factors affecting the decision to obtain a service auditor's report
- 2 Considerations in using a service auditor's report
- 3 Responsibilities of service auditors for special-purpose reports

When evaluating the controls at a service center by reviewing a third party report, the user auditor should consider the following factors:

- The service auditor's professional reputation
- The sufficiency of the controls procedures described in the report
- The scope of the compliance tests in relation to the user auditor's audit objectives
- The timing of the service auditor's report in relation to the period under audit

When reviewing and evaluating the controls at a service center, the service auditor is concerned with the same types of controls that would be encountered in a client's own data processing center. For a third-party report on the design of a system, the service auditor usually obtains information through discussion with service center personnel, review of documentation, and a walk-through of the procedures to confirm his understanding. For a third-party report on design and certain compliance tests, the system review procedures would be supplemented by tests of compliance with specific control procedures.

The following points should be considered in the review and evaluation of accounting control relative to processing of client financial data by a service center. The evidence for the evaluation by the user auditor comes from his own examination procedures and, if used, from the service auditor's report:

- Client provisions for liaison with the computer service center
- Procedures used to transmit instructions and data to the service center (Such

## 286 Auditing Service-Center-Produced Records

transmission should be restricted to authorized personnel, and summary and control reports prepared at the center should be delivered directly to the client personnel responsible for maintaining controls )

- Operating procedures employed at the service center
- Supervision exercised at the service center
- Security provisions over client data and files
- Backup and reconstruction provisions
- Methods for handling important conditions, such as—
  - Unmatched transactions (no master file records)
  - Control total or control count inconsistencies
  - Error corrections

### **EVIDENTIAL MATTER**

In an audit of a client that uses a service center, an auditor may use the same techniques used in an audit of an EDP system in which no service center is involved. For example, audit software may be used to examine file records, prepare independent control totals, and select records for confirmation or comparison with physical counts.

The nature of service center operations makes it important for the auditor to plan audit work well in advance of the audit date. Arrangements must be made for the retention of the necessary files and for the computer time required for his processing. If computer time will not be available at the service center, computer time at an alternate location must be obtained. Audit requirements should be discussed jointly with the client and the service center and confirmed in writing.

## **Time-sharing Computer Service Centers**

In time-sharing, a central computer system is used concurrently by several independent users. For example, some banks and savings and loan organizations use time-sharing services for on-line recording of savings transactions.

### **BASIC CHARACTERISTICS OF TIME-SHARING**

A time-sharing service center is organized differently from batch-oriented service centers. Each user has one or more input/output devices (terminals) connected to the central computer facility by data communication lines. The facilities of a telephone company, Western Union, or a private carrier are usually used for data communication. Users send data and instructions from their terminals as if they were the sole users of the computer. The central computer accepts data and instructions simultaneously from several users. By allotting each user a small segment of computer time, it services the users concurrently. The computer can usually return the requested output almost immediately. During peak periods of use, however, response time may become longer.

The user's data files are maintained at the central computer center. Instructions to the computer identify the files to be used. The system is designed to prevent one user from gaining unauthorized access to the files of another. There is a set of programs for each user's processing. These may be standard programs serving a number of users or, more commonly, specific programs prepared for each user. The data files, program files, and input/output devices are available immediately to the computer, so that processing can be performed at random as transactions or requests are entered.



These characteristics ensure the user immediate access to the computer for processing transactions or for obtaining data from files. Thus, the user may employ a service center and still have the quick access and response provided by an in-house computer

### **CONTROL FEATURES OF TIME-SHARING**

A time-sharing service center should establish controls and protections such as the following

- Protection against alteration or destruction of subscriber programs
- Control against unauthorized use of subscribers' proprietary programs
- Control against unauthorized access to subscribers' files
- Protection against loss or destruction of data files
- Provision for file reconstruction
- Provision for recovery from equipment failure
- Control against inaccurate transmission of data between the computer and the subscriber's terminal device

The protection of the user's data and programs is a function of the hardware controls and the controls in the operating system programs. The programs recognize the initiation of an action at a user's terminal device, and they screen and control all communications, perform editing, assign priority sequence, and bring into memory the appropriate user program. In some installations, one computer is used for storing the operating system programs, while a second computer is used for executing user programs. The operating system programs are usually very complex and are the responsibility of the computer center

Checking features should be written into the user's programs unless they are provided as part of the processing performed by the service center programs. These checks may include control totals for batches of transactions and for time periods, transaction listings, transaction numberings, and periodic file printouts and analyses

Again, it is important for the user to develop and maintain independent control totals for data being processed or stored in the center's files. These totals should be checked regularly against the comparable information furnished by the system during each processing routine. Another important control procedure involves regular review and testing of the data processing performed by the service center

When using a time-sharing computer service center, the user organization should ensure that the information required for audit purposes is readily available. If the requirements for an adequate audit (management) trail are met, the auditor should be able to perform most audit procedures by using data available from the file of user documents or from user computer files via terminal access

A time-sharing computer service center presents special problems for the auditor who wishes to review data processing thoroughly. A time-sharing system (particularly the system software) is highly complex. The system software for time-sharing applies to all the users of the center and affects the accuracy of the data processing. However, it is difficult for a user's auditor to obtain satisfaction about the computer center's time-sharing system software by direct examination because some centers maintain that allowing access to details of the system and to the physical computer facility is a serious breach of their control

Some time-sharing computer centers have retained accounting firms to examine the systems and to offer opinions about the organization and operation of the centers

## 288 Auditing Service-Center-Produced Records

Such reviews are particularly important if major processing tasks and file handling are performed by the time-sharing service. However, the reviews provide assurance only for as long as the system remains unchanged and is operated in the authorized fashion.

Thus, the internal controls for time-shared processing most important to the auditor are the controls exercised by the client over input data and the control totals, record counts, and so on provided by the user programs and the system programs. The audit trail of printouts in such a system should provide sufficient documentation for evaluating controls. Client records maintained at the service center may be tested by using printouts of the file or by running audit routines. When a computer program is used by the auditor for testing, it may be possible or even necessary to use the client's terminal devices.

### Summary

A number of different data processing services can be used to handle financial applications for user organizations that do not have their own computer facility. The use of these services involves some control considerations that are different from those encountered in dealing with an in-house data processing center.

If a client processes financial data at a service center, there are some unique audit problems. For example, while use of a service center usually improves the level of control regarding segregation of functions between record-keeping and custody of assets, it may weaken controls over processing. Time-sharing computer service centers have special control and operating characteristics that will influence the auditor's work.

The auditor should pay particular attention to the controls the client has over the processing performed by the service center. These controls are a key element in the auditor's evaluation of a client's internal accounting control.

It can be difficult for a service center to deal with the auditors engaged by the center's clients. To reduce this difficulty, the concept of a third-party review was developed. This involves using one auditor to review the center and provide an evaluation of the center's internal accounting control.

Appendix F presents a checklist to be used as the framework for developing a questionnaire to guide the conduct of a study and evaluation of the controls over the processing performed by a data processing service center.

---

# Control and Audit Considerations for Minicomputer and Microcomputer Systems

# 19

In the late 1970s, small, low-cost computer systems began to be used for business data processing applications. This chapter discusses audit concerns when a small computer system is used for processing of financial records.

## Minicomputer and Microcomputer: Definitions and Differences

The terms "minicomputer" and "microcomputer" refer to small systems, but the distinction is not precise. In general, the term "microcomputer" refers to a very small personal computer system. This computer system is housed in a cabinet the size of a portable typewriter, and it can be used together with a visual display screen for personal computing and for computer-assisted education. For business data processing, the basic computer and television monitor are supplemented by diskette devices for secondary storage and by a typewriter or low-cost printer for hard copy output. In addition, higher capacity disk units are sometimes included in the system. Microcomputers are being marketed by retail stores that also sell software for business data processing applications. Programming of microcomputers is generally in the simple BASIC language, although other languages are becoming available.

A minicomputer is also a small computer system, but generally one with more capabilities than a microcomputer. The minicomputers are essentially down-sized large computers with powerful processing capabilities but with limited input and output devices. A small minicomputer system and a large microcomputer system might appear to have similar input and output devices, but the minicomputer is more powerful and can generally be expanded to operate with more than one terminal while the microcomputer often supports only one input device.

Although many control and audit considerations are the same for both minicomputers and microcomputers, differences arise from the fact that minicomputers are likely to be operated by personnel trained for computer operations, and microcomputers are most frequently operated by user personnel who have other job responsibilities. That is, microcomputers will usually have no computer professionals associated with their day-to-day operations. Also, the minicomputer is generally provided space that defines a data processing function, a microcomputer may be on a desk or stand, and there will be no significant area dedicated to its operation.

## 290 Control and Audit for Minicomputer and Microcomputer Systems

Because of these distinctions, the first part of the chapter will describe audit and control for minicomputers or small business computers and the latter part of the chapter will cover microcomputers. Many minicomputers are used in such non-data-processing applications as process control. The chapter refers only to minicomputers in data processing. In this context, the minicomputer has the following characteristics:

- The computer is physically located in a user area. It is controlled and operated by a small staff associated with the user rather than a centralized EDP department.
- Only a limited number of applications are run; in fact, the system may be dedicated to one or two major applications.
- The computer system does not require special provisions for air conditioning, humidity control, or electrical power, and it may be difficult to limit physical access to the computer.
- A limited number of people are involved in data processing operations using the system, so it is difficult to achieve an effective segregation of functions and responsibilities.
- Processing is frequently by immediate entry, using one or more entry terminals and immediate processing. Terminals that access the system are physically close to the computer and are directly connected to it.
- The computer system has limited secondary storage capabilities for data files, which places restrictions on the amount of data that can be retained and on the length of time data can be stored.
- Software is frequently purchased rather than developed in house.
- Printing capabilities are limited, so the system will not be used for applications that involve heavy print volumes.

### **Minicomputers and Distributed Data Processing**

A significant concept implemented by some organizations is the distribution of data processing capabilities, including hardware, to user locations. This dispersal can take a variety of forms. Each data processing location can have independent minicomputer hardware with its own files. Or, each data processing minicomputer can operate independently for some local processing as part of a network connected by data communications, with computers in the network sending and receiving data from other minicomputers in the network. Finally, a central computer can maintain a central database, with each minicomputer in the network accessing and updating the central database as well as maintaining local data files.

These distributed processing system designs are intended to provide a more responsive data processing system and to reduce the risks inherent in large, single-installation systems. From an audit standpoint, the distribution of processing through a network of minicomputers is likely to have many of the internal control characteristics of large systems because of the control necessary for operations. However, the option of having independent hardware that is not connected by a network may mean that each location is independent with respect to controls. The discussion in this chapter applies to the independent minicomputer.

## Effects of Minicomputers on EDP

### SPECIALIZATION

Generally, a minicomputer is used for only a small number of applications. This limited use is likely to foster a trend away from the use of technical specialists in data processing and from segregation of technical duties. Most minicomputer installations will be supported by a relatively small staff, who will handle a wide range of data processing tasks—circumstances likely to create problems in maintaining segregation of functions.

### HARDWARE AND SOFTWARE STANDARDS

Hardware and software standards for computers have been both de facto and explicit. De facto standards have resulted because of the dominance of IBM in the large-scale data processing computer market. Explicit standards have been established through cooperative standards programs, such as the American National Standards Institute or the International Standards Organization.

These two factors, as well as the existence of only a small number of computer manufacturers, have resulted in a fairly high level of standardization for larger systems. In the minicomputer market, however, a large number of manufacturers maintain a high level of innovation, leading to a lack of compatibility and standardization.

### PROGRAMMING LANGUAGES

COBOL is the dominant, standard language for business processing, in fact, the language was created for that. However, many minicomputer systems do not offer COBOL, and minicomputer suppliers have often implemented different languages for use on their systems.

### LOGGING AND AUDIT TRAILS

The recording of all transactions in a system log, commonplace in medium- and large-scale computer systems, is often not done in the minicomputer environment. Minicomputers do not have the wide range of peripheral devices required to record and retain the volume of information necessary to provide a magnetic media audit trail. As a result, the audit trails that are available on magnetic media are limited in that they are only maintained for a short time. Alternatives are to provide adequate audit trails on printed output or to add logging capabilities to the minicomputer.

### FILE/PROGRAM BACKUP

File and program backup is usually established by creating a duplicate copy of a file and storing the copy in some off-premises location. In larger systems the file or program is typically written on magnetic tape that serves as the off-site medium for retaining the backup copies. However, minicomputer systems may have little or no magnetic tape capability. Therefore, low-cost backup and recovery is an important consideration in minicomputer system design. One alternative is diskette backup storage. It is most efficiently done with two disk drives: the original file on the first drive is copied onto a second diskette, which is stored off-site. Low-cost tape cassettes can also be used for backup storage.

### **HARDWARE VENDOR SUPPORT**

The relations between the users and vendors of data processing hardware are different in a minicomputer environment from those in a large computer system. There is little in common among the various kinds of minicomputers, and there are no standards for the development of mini-based application systems. Therefore, it is difficult for a user to switch from one minicomputer vendor to another.

Because of the low cost of minicomputer hardware, vendors provide a low level of technical support for users. The vendor-supplied systems engineers and technicians available to advise users of medium-scale and large-scale systems are not available for minicomputers.

Vendors also find it impossible to maintain a complete, on-call network of technicians to repair and maintain minicomputer hardware. On-site preventative maintenance is not usually performed. There is also a trend toward making the user more self-sufficient when it comes to repairs. Some minicomputers come with spare parts, such as circuit boards, and instructions that guide users in making their own repairs.

## **Audit Concerns and Audit Planning When Minicomputers Are Used for Financial Records**

The impact of minicomputers in EDP also affects auditing. Primary areas of concern are audit planning and continuation provisions. Control evaluation and testing considerations are described later in the chapter.

### **AUDIT PLANNING**

In planning the audit involving a minicomputer installation (particularly in a large company), the auditor may find that several machines have been installed in various locations. Thus, the auditor must select both the minicomputer locations and the applications to be reviewed.

In planning the work, the auditor will need to give special consideration to the timing of activities and tests. There are severe limits on the number of data records that can be retained on magnetic media within a minicomputer system. Audit trails and other supporting information on magnetic media may only be retained for a limited period of time. The auditor who plans to test some specific aspect of the system, or who plans to test transactions that took place on some specific date, must plan the audit work so that it takes place when the supporting information is available.

Another planning consideration is the advance preparation that may be necessary in order to apply audit procedures using audit software. Programs or packages for use in audit activities must be identified or written and tested.

### **CONTINUATION PROVISIONS WITH MINICOMPUTERS**

If continuation provisions are included in the scope of the engagement, the auditor will have to pay special attention to backup and recovery provisions. Hardware backup involves different considerations than would occur for a large computer system. Because of the lack of standardization, differing minicomputers generally cannot support one another. However, minicomputer hardware is readily available and could be acquired quickly in an emergency.

In addition, many organizations that have a number of minicomputers from the same vendor may find it easier to provide backup. Functions may be shifted among the minicomputers, making hardware available for vital processing. Should one system fail, another may be able to assume part of the load.

## Control and Evaluation in Minicomputer Systems

In reviewing the effects of the minicomputer on the auditor's study and evaluation of internal accounting control, it is convenient to use the two categories of general controls and application controls as the basis for discussion. The discussion explains how the two types of controls may be affected by the use of a minicomputer system. The controls are taken from the AICPA Audit and Accounting Guide, *The Auditor's Study and Evaluation of Internal Control in EDP Systems*, chapter 3, and the AICPA Computer Services Guideline, *Audit and Control Considerations in a Minicomputer Environment*.

### GENERAL CONTROLS IN THE USE OF MINICOMPUTER SYSTEMS

*Segregation of functions between the EDP department and users* In a small organization using a minicomputer system, the users may perform the EDP functions, and no effective segregation of functions is possible.

The auditor should make management aware of the risk inherent in failing to provide a segregation of functions. With the auditor's assistance, management may design as much separation of functions as possible and identify and implement alternative controls or compensating controls.

One method for establishing some segregation of functions among employees in the user department is to make an employee who does not enter or process data responsible for maintaining a control log. The purpose of the control log is to record control information, to balance computer control totals with other control information, and to provide a mechanism for alerting management to situations requiring investigation and corrective action. Examples are—

- Transaction balances by type of transaction, such as cash receipts, cash disbursements, or inventory receipts, and balancing with totals developed by other personnel who do not operate the minicomputer or by other processing operators
- Master file control totals, such as dollar values and number of records, for financial files. (The beginning balance plus transactions affecting balances yield an ending balance that is compared with control totals for a file obtained periodically by a special file control total run.)
- Control comparison based on characteristics of the business to identify possible errors
  - Number of orders times average expected order size compared with the actual order transaction total
  - Estimated cost of goods sold computed by using ratios compared with actual cost of goods sold based on transactions

Because of the lack of segregation of functions between data processing and the user department, there should be emphasis on dual controls and reviews of transactions affecting assets. Examples of asset controls to be considered are these:

## 294 Control and Audit for Minicomputer and Microcomputer Systems

- All check signing should be subject to an effective dual control. Individuals authorized to sign checks should be provided with all the documentation they need to authenticate the propriety of the transaction.
- Physical control over the movement of assets should be maintained. In particular, a company should emphasize control over assets that are portable and have a reasonably high monetary value.
- The application of cash or other payments against open accounts receivable should be subject to dual control.

*Provision for general authorization over the execution of transactions (prohibiting the EDP department from initiating or authorizing transactions)* Proper implementation of this control depends on providing a basic segregation of functions. In any case, management should understand and approve the types of transactions to be processed by the EDP system and the rules that have been established to govern that processing.

The auditor should be aware of the processing rules that have been established for the transaction types that are to be handled, and any exception should be carefully reviewed. Unusual transactions or questionable processing should be reviewed with management. Since programs perform processing the same way given the same conditions, any exception may indicate a pattern of similar exceptions.

*Segregation of functions within the EDP department* Effective segregation of functions within EDP is not likely to exist in a minicomputer environment. As a minimum, it would be desirable to have a segregation between data entry and processing. In the absence of such segregation, the auditor must place increased reliance on segregation of functions between EDP duties and user authorization for transactions. If they have been effectively implemented, they offset the weakness of segregation of functions within EDP. However, if those controls are also weak, the auditor cannot rely on the output produced by the data processing system. In such cases, the audit program must emphasize substantive tests.

Many installations do not write programs but rather purchase packaged software. This practice results in segregation between the programming and operations functions. When development is in-house, the programming and operations functions should be segregated, but in a minicomputer environment, they often are not.

*The procedures for system design, including the acquisition of software packages, should require active participation by representatives of the users and, as appropriate, the accounting department and internal auditors.* In a minicomputer system, it is likely that the choice of software will be influenced by the hardware that is used. Therefore, the main area of user involvement is likely to be in the selection of a complete system, including both hardware and software. If the users understand a system and participate in its design or selection, they are more likely to make the system work and to make the controls within the system effective. If users do not have a sense of direct participation, they are likely to be less interested in making controls effective.

The extent of user involvement can usually be determined by talking to the users. If their reaction to the system is one of resignation, skepticism, or anger, they have probably committed themselves to fighting the system, which will hamper the effectiveness of any data processing effort. In such cases, the auditor should take a more critical view of user controls.



*Each system should have written specifications that are reviewed and approved by an appropriate level of management and applicable user departments* This type of control is fundamental. It is an extension of good business practice and fairly easy to implement. It is specifically designed to promote user and management involvement in data processing activities. In smaller systems, the implementation of this control may represent a problem. Adequate documentation should be a requirement in the purchase of all application software, but documentation supplied in support of some purchased minicomputer application packages may be poor.

The effect of a weakness in this control is much the same as that of a weakness in the general control regarding user participation. The auditor should determine the extent of the review and approval process associated with the design or selection of new systems. There should be appropriate evidence that users and management understood the system they approved.

*System testing should be a joint effort of users and EDP personnel and should include both the manual and computerized phases of the system* System testing is of particular importance in the initial conversion from a manual system to an electronic data processing system. It is likely to be particularly important in a minicomputer system. Use of the minicomputer may be the first application of data processing within a particular organization. In conducting application system tests, it is important to include unusually large or complex transactions, since they will often cause a system to fail.

Systems testing of minicomputers should also include transactions to test the following:

1. Fields are provided within the records that can accommodate the numerical amounts involved.
2. The contents of fields are specified so that fields to contain numbers always contain numbers and those to contain alphabetic data contain alphabetic data.
3. Transactions that do not fully offset one another are detected and noted as errors. For example, in a transaction to establish an account receivable balance, the corresponding entry that offsets the receivable amount should be for the same amount.
4. Transactions that involve splitting an entry within a system are handled properly, for example, a cash payment that covers more than one open invoice.
5. Corrections, particularly corrections to be made in the master file, are processed properly.

In the absence of an effective level of user system testing, the auditor should consider performing audit test procedures from these five broad test areas. A simple set of test transactions may allow the auditor to detect major system flaws.

*Final approval should be obtained prior to placing a new system into operation* This is another control designed to increase user and management involvement in the system. A weakness in this area of control would have much the same effect as the weakness described for the user participation in systems design.

*All master file and transaction file conversion should be controlled to prevent unauthorized changes and to provide accurate and complete results* This control is important in systems requiring conversion of files from one format or medium to another. In the minicomputer environment, manual files are usually converted to computer files on the

## 296 Control and Audit for Minicomputer and Microcomputer Systems

minicomputer Conversions performed in the minicomputer environment should be subject to adequate control. Such controls are designed to ensure that the data are neither distorted nor lost during the conversion process.

The auditor should determine the extent of the file conversions that are taking place. If they are significant, an evaluation of the control becomes more important. The absence of effective control is a serious system failure, and it might result in the auditor's placing little reliance on results produced by processing that uses the contents of the converted files.

*After a new system has been placed in operation, all program changes should be approved before implementation to determine whether they have been authorized, tested, and documented.* The effect of a minicomputer system on the control over program changes is mixed. In some applications, the control may be improved, in others, it may be considerably weakened. The key factor is whether the user has the opportunity to change the software.

In many minicomputer systems, the user has no direct involvement in the preparation or maintenance of the software because application packages are purchased from an outside source. Normally, as part of the purchase agreement, maintenance is provided by the organization that supplied the package. In these situations, the user will not have the capability to make a change in the software. This is true for turnkey systems for which users are not provided a source code. In such cases, the basic intent of the control is clearly satisfied, and the auditor may place reliance on it and its impact on the overall reliability of the system if vendor changes are adequately tested.

In some cases, the minicomputer users may be involved in maintaining their own software. In a small computer system, there is very little control over such changes. Program change control requires a level of discipline that is rare in small data processing organizations, and, in such an environment, the auditor would not be able to place reliance on it.

*Management should require various levels of documentation and establish formal procedures to define the system at appropriate levels of detail.* This control, in the minicomputer environment, is very similar to the requirement for written systems specifications. Good documentation of application systems is important to users, management, and auditors; any applications that do not have good documentation should not be considered for acquisition.

A small data processing installation that develops its own application system may not have good documentation discipline. There may be a lack of management support for documentation, if so, data processing may not spend the effort required to prepare it. A weakness in this control need not necessarily decrease the auditor's reliance on a system, but it does make audit procedures more difficult and time consuming. In order to review the system, the auditor may be forced to produce minimum levels of documentation.

*The control features inherent in the computer hardware, operating system, and other supporting software should be utilized to the maximum possible extent to provide control over operations and to detect and report hardware malfunctions.* This is likely to be a weak area in most minicomputer systems. The relatively unsophisticated operating systems available in this environment generally do not provide the controls that are more familiar in large-scale systems. For example, for most minicomputers there are no

header or trailer labels on magnetic files, so the system does not perform label checking. Provision for password protection varies. Little, if any, logging is carried out by the system.

The hardware is very reliable, but the circuitry usually provides little redundancy checking. For example, parity checking is seldom used, and the reading of input is almost never subject to dual read and check. However, these weaknesses probably do not represent a significant problem. Computer hardware is, by and large, highly reliable, weaknesses in hardware control will have little or no effect on the processing of applications. Good application controls can probably more than offset any problems in minicomputer hardware or software controls.

*Systems software should be subjected to the same control procedures as those applied to installation of and changes to application programs.* In a minicomputer system, this control area generally will not represent a problem. The operating systems are less complex than large-scale systems and are subject to few changes.

In most minicomputer systems, a complete replacement of the operating system is much more likely than modification by a user. If the auditor has reason to doubt the integrity of the minicomputer system software in a given situation, he should obtain and use an audit copy of the software, which can be obtained from the manufacturer or vendor.

*Access to program documentation should be limited to those persons who require it in the performance of their duties.* This is a good rule, but it can be difficult to enforce when the data processing group is small and the division of duties is not enforced. However, the number of people who can understand and use documentation is generally limited. If personnel with access to documentation have the skill to make changes, a weakness in control in this area may result in undocumented and unauthorized changes. Therefore, the auditor may not be able to place reliance on documentation as accurately reflecting the application system.

*Access to data files and programs should be limited to those individuals authorized to process or maintain particular systems.* Access to data files and programs in minicomputers may be based on passwords issued to authorized users. The use of passwords may not be enforced to the extent that is desirable because of limited staff and the lack of clear division of duties. This may result in unauthorized changes in data records, therefore, the auditor may need to increase substantive tests on the data files.

*Access to computer hardware should be limited to authorized individuals.* Generally speaking, this is a good management practice. The computer is a valuable business asset, so access to it should be limited. Further, preventing unauthorized access will reduce the possibility of unauthorized use of information or unauthorized changes in the system. However, minicomputers are frequently placed in locations without physical access control. A weakness in this control might cause the auditor to place less reliance on the data records stored in the system.

*A control function should be responsible for receiving all data to be processed, for ensuring that all data are recorded, for following up on errors detected during processing to see that they are corrected and resubmitted by the proper party, and for verifying*

## 298 Control and Audit for Minicomputer and Microcomputer Systems

*the proper distribution of output* A separate control function in a minicomputer system environment may not be feasible, however, control duties should be identified and assigned to individuals who do not perform the processing. For example, clerical or accounting personnel who do not perform data processing may perform control functions. If the assignment of control duties is not adequate, the auditor must place increased reliance on substantive testing.

*A written manual of systems and procedures should be prepared for all computer operations and should provide for management's general or specific authorizations to process transactions* This control is really another form of documentation. A control weakness in this area would not be critical from the audit standpoint, but a complete lack of control would be evidence of a poor operation, which would tend to decrease auditor reliance on the results of processing.

*Internal auditors or some other independent group within an organization should review and evaluate proposed systems at critical stages of development* This control may not exist if the minicomputer is in a small organization. As a result, the independent auditor should place increased reliance on the results of his own work in evaluating the controls and on substantive tests.

*On a continuing basis, internal auditors or some other independent group within an organization should review and test computer processing activities* If there is no internal audit function, this control may be exercised by management and user groups. One approach is to plan and conduct formal meetings of management, users, and EDP on a regular basis to review problems, discuss system modifications, and plan future developments. Such meetings can provide a reasonable substitute for the formal development control that exists in a larger installation.

### PHYSICAL SECURITY FOR MINICOMPUTERS

Auditors do not usually consider physical security as a general control. Some physical security considerations, however, may have an effect on the overall evaluation of control.

*The Auditor's Study and Evaluation of Internal Control in EDP Systems* mentions four basic areas of physical security that should be of concern to the auditor. Two of these areas, environmental controls and protection against fire and other hazards, are not necessarily applicable to a minicomputer system, because most minicomputers do not require special environmental controls. Further, minicomputers are relatively inexpensive and easily replaceable, so protection against fire or other physical hazards is not a major concern.

Two other aspects of physical security may be important to the auditor in a minicomputer system. One is the off-premises storage and retention of files and programs. In a minicomputer system, the value of the data files will usually exceed that of the physical hardware. Therefore, the auditor should evaluate the system to determine whether proper provisions have been made for off-premises storage of important files and programs.

The other aspect of physical security in a minicomputer environment is the need to provide business interruption data processing insurance to protect the installation.

against unusually large losses caused by a loss of computing capacity. The audit guide suggests that a material uninsured risk might require financial statement disclosure.

## **Application Controls in the Use of Minicomputer Systems**

Most of the application controls that are outlined in the audit and accounting guide and described in this book are as important in a minicomputer system as they are in a larger system. As a result, this section deals only with those application controls that may require special consideration when evaluating controls in a minicomputer system. The controls listed below are taken from *The Auditor's Study and Evaluation of Internal Control in EDP Systems*, chapter 4.

*Conversion of data into machine-readable form should be controlled.* In the minicomputer environment, there is likely to be a reduction in the amount of data to be converted into machine-readable form. The trend is for the original data to be entered directly into the minicomputer system through a terminal, so that data conversion is usually not a problem. However, since the data are being recorded directly, increased importance is placed on control over original input data.

*Movement of data between one processing step and another, or between departments, should be controlled.* Although this control is still required, it is likely to be of less importance in a minicomputer environment. There is much less movement of data between departments, so the need to provide control over that movement is reduced.

*Controls should prevent processing the wrong file, detect errors in file manipulation, and highlight operator-caused errors.* In most minicomputer systems, this control cannot be implemented to the extent that it is implemented in larger systems. Minicomputers do not have the hardware and software provisions for checking file labels or for preparing system activity logs used to record operator error. There must be a reliance on external labeling of disks and tapes and on storage and use procedures.

*Limit and reasonableness checks should be incorporated within programs.* Many minicomputer systems use application software packages produced by software vendors. As a result, the limit and reasonableness checks incorporated in the application package vary with the sensitivity of the vendor to control requirements. In most cases, it will be impossible for the user to retrofit the kinds of checks that should be included. Therefore, the existence of adequate limit and reasonableness checks should be a major factor in evaluating alternatives when selecting an application package.

*Systems output should be distributed only to authorized users.* Because minicomputers are normally used in smaller organizations or small divisions of larger organizations, a formal control over output distribution is usually not required. In most cases, the data processing people who perform the distribution function will be thoroughly familiar with the authorized employees who should receive copies of the output, which gives them a high level of distribution control.

## Performing Audit Tests

Compliance with minicomputer system controls described by the documentation and by client personnel can be tested by observation and by examination of evidence of performance, such as documents, reports, signatures, and initials. Because of the relative simplicity of the computer processing environment and the applications performed, an auditor can walk through processing and control procedures and observe performance.

In some cases, it may be desirable to use audit procedures to test the processing within the computer, which involves procedures to test application programs and system software. However, audit procedures for substantive testing are probably more efficient in most cases.

Substantive tests can be performed on the data produced by the processing applications. The results consist of transactions and master files. In both cases, there may be printed output that may be used instead of machine-readable records. However, it is frequently desirable and efficient to perform substantive tests using the client's minicomputer. Generalized audit software used in testing in larger systems may not function on a minicomputer system. Also, other software tools developed to assist the auditor are not designed to operate on minicomputer hardware. The auditor may use the following:

- Simple audit programs written by the auditor (This is frequently feasible because of the simple environment, for example, some tasks may use standard auditor-designed program modules to foot files.)
- Client query software or utilities, if available
- Report-writer programs, if available

Because a minicomputer system is generally less controlled than a larger EDP system, the errors and exceptions detected during testing need to be fully explored. Any exception that can be a result of an error or irregularity in program logic is especially significant because it will be repeated consistently.

## Control and Audit When Microcomputers Are Used for Data Processing

Many microcomputers in organizations are personal computers used in planning, accounting functions, decision analysis, maintaining local files, word processing, and statistical analysis. These applications sometimes have little relevance to the financial statements. However, microcomputer systems may perform specialized data processing in large organizations and perform a wide range of data processing in smaller organizations. For example, a small business may use the microcomputer for inventory, accounts receivable, payroll, purchasing, and general ledger and also use it for correspondence and other word processing. The computer takes the place of a book-keeping machine or manual recording. Microcomputers are used in organizations of all sizes for electronic worksheet applications supporting a wide variety of accounting functions.

The microcomputer for data processing will often have the following characteristics:

- There is no separate data processing staff trained in computing. The training of the personnel using the microcomputer is oriented toward the applications they process.

- The computer and its peripherals are on a desk or stand rather than a computer area or computer center
- There is no physical security for the computer
- There may be a rigid (hard) disk for system software and programs in use, auxiliary storage is probably on flexible (floppy) diskettes

The lack of personnel trained in computers means that the data processing procedures to be followed should be well documented and the personnel should be given appropriate training. One area of difficulty is the backing up of files. This is a separate task that may not be done unless the organizational procedures and controls are properly supervised. The backup diskettes should be properly labeled and then stored in a separate, secure location. Use of backup diskettes to restore files should be controlled by supervisory personnel. A simple library function may be assigned as a part-time duty to a person having other compatible duties.

Physical security may be enhanced by log-on procedures. Also, the microcomputer may be disabled for data processing by removing critical software each night and restoring it each morning.

The data processing applications should be designed with printouts at frequent intervals, and there should be control totals that are traced to the general ledger accounts.

The auditor can ask for printouts of the files being maintained on the system, also, the auditor can use retrieval packages if one is available. In addition, the auditor may write a program in a simple language available on the microcomputer.

## Summary

Both minicomputers and smaller microcomputers are being used for business data processing. For the most part, the audit problems that have been encountered with the use of minicomputers do not require unique solutions or the development of new audit techniques. Rather, currently available tools and techniques can be applied to a mini-computer system, including each of the general controls for EDP and certain application controls.





---

## **Audit Staff Selection, Organization, and Training for EDP Auditing**

# 20

An auditor cannot adequately audit a system that he does not understand, nor can the auditor make adequate use of computers in audit tasks unless he possesses a reasonable knowledge of EDP. The level of EDP knowledge and proficiency required by an individual auditor to meet these needs depends both on the complexity of the system being audited and the degree of EDP expertise available to the audit team. The proficiency requirements for the internal EDP auditor and the independent EDP audit specialist are similar in many ways; however, this chapter is directed to the independent auditor. This chapter describes alternative approaches to the organizational aspects of EDP auditing, defines technical proficiency requirements, and describes approaches to staff training for EDP auditing.

### **Options in Organization of EDP Auditing Function in a CPA Firm**

An auditing firm has the following options in organizing and staffing its EDP audit activities.

- No internal specialization
- Outside EDP audit specialists
- EDP audit specialists on the audit staff
- Management advisory services EDP consultants serving as EDP audit specialists

Each of these options has advantages and disadvantages. A firm may follow a mixed strategy, employing different options for different offices or for different clients. The underlying concept for all the available options is that the field auditor having a reasonable expertise in EDP should have EDP audit specialists available for consultation and assistance when systems are complex or when specialized audit techniques are indicated. Supervision and review are a major consideration for all options. The supervising auditors and the managers and partners who review and approve the audit plan and the audit working papers should, therefore, have adequate technical knowledge for review purposes. The hierarchical nature of supervision and review means that the firm should establish suitable degrees of technical proficiency for each level from the field audit staff up to the partner in charge of the audit.

### **NO INTERNAL SPECIALIZATION**

Many firms have defined areas of staff specialization, while others prefer to limit specialization and expect their staff to be able to handle any assignment the firm is willing to accept. This latter approach has the advantage of flexibility in staff assignments and in supervisory and partner reviews. It has the disadvantage of limiting the firm's ability to handle jobs requiring in-depth expertise. For example, a firm organized in this way may have difficulty auditing clients equipped with complex EDP systems.

### **OUTSIDE EDP AUDIT SPECIALISTS**

A firm that has a staff with moderate EDP proficiency and a small number of clients with complex computer systems may obtain adequate EDP proficiency by using another CPA firm to provide technical support. The specialist firm assists in the audit plan, in field work, and in field work supervision. They also advise the partner with respect to technical questions that arise during the review.

### **EDP AUDIT SPECIALISTS ON THE AUDIT STAFF**

A firm may choose to develop specialized expertise on the audit staff. In this option, all staff auditors are expected to have a moderate EDP knowledge, but some auditors are given additional training to provide them with a higher level of proficiency in EDP and EDP auditing methods. One or more of these specialists is assigned to each audit in which the client has a complex data processing system.

The specialization at the field level is carried through the hierarchy of the firm. Some supervisors, managers, and partners also have specialized knowledge, and they plan, supervise, and review the work of the EDP audit specialists in the field.

The principal advantage in having EDP audit specialists is the high proficiency level that can be applied to an audit and to the related supervision and review process. The problems of developing and maintaining a specialization within the audit staff and of establishing a career path for the EDP audit specialists are disadvantages.

Developing and maintaining an EDP audit specialization can be difficult because the relatively small number of qualified staff in the firm or office will limit flexibility in assignments, and employee turnover can have a significant effect on the firm's ability to staff EDP audits.

Career paths for EDP auditors may be difficult to define. Maintaining a specialized competence may interfere with the development of general audit skills that are required for advancement in the firm.

### **MANAGEMENT ADVISORY SERVICES EDP CONSULTANTS SERVING AS EDP AUDIT SPECIALISTS**

EDP consultants assigned to management advisory services may be used as EDP audit specialists. EDP consultants maintain technical skills as part of their MAS professional development, and they usually have a career path within MAS.

There are disadvantages, however. MAS personnel may not have an audit orientation. Some firms feel that the qualities that make successful MAS consultants differ from those that make a successful auditor.

## Technical Proficiency for Auditing Computer-Processed Records

Because of the pervasive use of computers in data processing and the generally accepted auditing standard of technical proficiency, all auditors are required to have a moderate level of data processing and EDP auditing knowledge

### STANDARDS FOR TECHNICAL PROFICIENCY

The pertinent AICPA standards for EDP expertise on the field audit staff and in the supervision of their work, as described in SAS 1, are

The examination is to be performed by a person or persons having adequate training and proficiency as an auditor (sec 210 1)

The work is to be adequately planned and assistants, if any, are to be properly supervised (sec 311 1)

These standards emphasize audit capability. A trained auditor can audit an automobile manufacturer without being an automotive engineer or an airframe manufacturer without being an expert in aeronautics. Data processing is different, however. The computer data processing system is often an essential part of the accounting system that the auditor must understand in order to study and evaluate internal accounting control and perform compliance and substantive tests.

### EDP KNOWLEDGE FOR ALL AUDITORS

The knowledge level for all certified public accountants just entering the profession was defined in 1966 in a common-body-of-knowledge study that recommended three major areas of knowledge with respect to computers<sup>1</sup>

- 1 *Hardware/software functions* "The beginning CPA should have basic knowledge of at least one computer system. This implies a knowledge of the functions of the component parts, of the general capabilities of the system, and of the more universal terms associated with the computer."
- 2 *Computer application processing* "He should be able to chart or diagram an information system (application) of modest complexity. This means that he should be able to comprehend the procedural steps in a system (an application) and utilize basic diagram symbols that describe the system clearly and precisely."
- 3 *Application development and implementation* "He should have a working knowledge of at least one computer language. With an understanding of a programming language together with his overall knowledge of information systems, the beginning CPA should be in a position to design a simple information system (application), program it, and proceed to debugging and testing."

---

<sup>1</sup> Robert H. Roy and James H. MacNeill *Horizons for a Profession* (New York: AICPA, 1966) p. 213

## 306 Audit Staff Selection, Organization, and Training for EDP Auditing

In 1971, a group from an AICPA subcommittee on computer education proposed an expansion of the common body recommendations.<sup>2</sup> The 1971 suggestions were updated in a 1975 *Journal of Accountancy* article,<sup>3</sup> which suggests eight minimum knowledge requirements for a general staff auditor

- 1 A basic understanding of computer systems including equipment components and their general capabilities
- 2 A basic understanding of widely installed computer operating systems and software
- 3 A general familiarity with file processing techniques and data structures
- 4 Sufficient working knowledge of computer audit software to use existing standardized audit packages
- 5 Ability to review and interpret systems documentation including flowcharts and record identification
- 6 Sufficient working knowledge of basic EDP controls to
  - a Identify and evaluate the controls in effect in a client's installation
  - b Determine the extent to which such controls should be tested and to evaluate the results of such tests (although not necessarily to execute such tests)
- 7 Sufficient knowledge of EDP systems to develop the audit plan and supervise its execution
- 8 A general familiarity with the dynamics involved in developing and modifying programs and processing systems

The changes in hardware, software, data processing systems, and audit techniques from 1975 to 1980 suggest additional requirements for all auditors

- General familiarity with database management functions and database software
- Sufficient working knowledge of the data administration function to be able to identify and evaluate the data control role of the data administrator
- General familiarity with security and continuation provisions

### COMPUTER AUDIT SPECIALISTS

In contrast to the general knowledge expected of all auditors, the computer audit specialist has a significantly higher level of EDP expertise and should be able to perform tasks such as these

- Provide technical support for the typical uses of generalized audit software
- Perform unique or complex tasks with generalized audit software
- Apply other computer-assisted audit techniques as appropriate
- Perform reviews where EDP technical ability is required, for example, reviews of system software controls
- Write or review computer programs
- Perform technical interface with client data processing personnel

---

<sup>2</sup> Richard W. Cutting, Richard J. Guilinan, Fred L. Lilly, Jr., and John F. Mullarkey, "Technical Proficiency for Auditing Computer Processed Accounting Records," *Journal of Accountancy* (October 1971) 76-78.

<sup>3</sup> Elise G. Jancura, "Technical Proficiency for Auditing Computer Processed Accounting Records," *Journal of Accountancy* (October 1975) 46-59.

- Evaluate and test security and continuation provisions
- Evaluate and test database management system operations and controls

Two major issues regarding the use of computer audit specialists are their selection and training. The issue of selection involves whether it is better to recruit auditors and train them in EDP or to recruit EDP specialists and train them in auditing. Both methods are advocated and used. The major argument for starting with data processing expertise is that this knowledge is more difficult for the CPA firm to provide. The major arguments for starting with auditors are that the basic specialist task is, in fact, auditing and that the motivations and rewards of the job may be more appealing to an auditor than to a data processing technician.

## Academic Preparation for EDP Auditing

The common-body-of-knowledge study described a minimum level of academic preparation in computer data processing for auditors. Later recommendations for more extensive academic preparation for auditors suggest that, in the future, entering CPAs will have more academic preparation in computer data processing.

In 1978, a task force defined a 150-semester-hour educational program for entry into the accounting profession, which represents the equivalent of a five-year professional degree.<sup>4</sup> Computer requirements include the following:

- General education. Introduction to Computers (three semester hours) includes learning a programming language.
- Accounting education. Computers and Information Systems (six semester hours) covers analysis and design of applications, additional computer programming, and control procedures in EDP systems.
- Auditing education. The required six semester hours of auditing includes the use of computers and the auditing of computerized systems.

Less rigorous four-year academic programs should generally provide at least these requirements:

Computer data processing

Programming (COBOL preferred)

Introduction to EDP auditing (as part of the auditing course)

Many schools offer additional, elective courses:

Auditing and EDP

Analysis and design of information systems

Management information systems

Data structures

Database management systems

Management and control of information systems

---

<sup>4</sup> "Task Force on the Report of the Committee on Education and Experience Requirements for CPAs," *Journal of Accountancy* (March 1979) 121-28.

## **Sources of Continued Staff Training**

Auditors need to keep current in the rapidly changing computer field. These training needs can be met by a variety of continuing education methods.

### **CONTINUING PROFESSIONAL EDUCATION COURSES OFFERED BY THE AICPA AND STATE SOCIETIES**

The AICPA offers a number of courses, which are administered by state CPA societies,<sup>5</sup> and some societies may have additional offerings. Typical course titles relevant to EDP and EDP audit training are these:

- Developing computer-based general ledger systems
- How to help your client evaluate his EDP needs
- How to help your client select the specific EDP system
- Computer controls and auditing
- Introduction to EDP concepts and controls
- Hands-on auditing of small business computers
- The effects of EDP on the auditor's study and evaluation of internal control
- EDP feasibility study
- Managing the human element in EDP

### **COURSES OFFERED IN THE BUSINESS ENVIRONMENT**

Computer manufacturers are a major source of computer training. They usually offer general orientation courses as well as courses in programming, system software, and other technical areas. In general, the content of the courses and the quality of the instruction have been good.

### **COURSES BY COLLEGES AND UNIVERSITIES**

Most colleges and universities have computer science, management information systems, and data processing courses as part of degree programs as well as continuing education offerings that may be attended days or evenings.

### **COURSES AT LOCAL TECHNICAL SCHOOLS**

In most metropolitan areas, technical schools offer EDP courses. Evaluation of this training source should consider the availability of equipment for student use, the qualifications of instructors, the outline of the course, and the comments and recommendations of persons who have completed the training.

### **SEMINARS AND SHORT COURSES**

Many private companies, educational institutions, and professional organizations offer seminars and short courses on EDP topics.

---

<sup>5</sup> Practitioners should check the AICPA Continuing Professional Educational Catalogue, published annually for titles of currently available courses.

**SELF-INSTRUCTION AND PROGRAMMED LEARNING**

The general principles of computer data processing and many elements of programming can be learned through the number of programmed self-study courses that are available. Several computer manufacturers use the programmed learning method. Compilation and debugging experience in programming can be obtained by the use of time-sharing. In addition, home study computer courses are offered by several institutions.

**IN-FIRM COURSES**

Several organizations sell or rent materials, often including video cassettes or film, designed for individual or small-group training in a CPA firm's own offices.





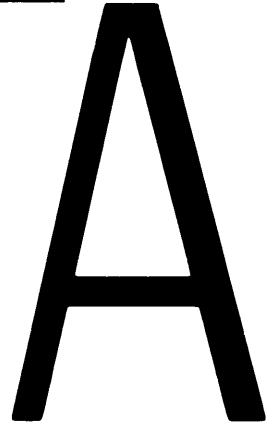
---

# Appendixes



---

# Questionnaire for the Evaluation of Internal Control in Computer Data Processing



This appendix contains an illustrative questionnaire for use as one method of obtaining information on internal control in computer data processing

The questionnaire is divided into four major parts

- 1 Background information on the hardware, software, and computer data processing organization
- 2 General controls in the computer data processing installation
- 3 Provisions for assuring continuation of processing in the event that some or all of the processing capabilities are destroyed or made unavailable
- 4 Controls in each computer data processing application being reviewed

This arrangement reflects the fact that the hardware, software, organization, general controls, and continuation provisions of the installation provide the environment in which individual applications are processed. This environment must be understood before the controls associated with individual applications can be evaluated.

The review of a computer processing application such as accounts receivable or inventory accounting should be carried out in the context of the entire processing cycle, including both computer and noncomputer processing and controls. The firm's internal review questionnaire (or other method used to obtain information) should cover both computer and noncomputer procedures and controls, the illustrative application questionnaire is structured to provide only those questions related to computer processing.

The scope of the illustrative internal control review questionnaire is limited to items that may affect the processing of financial records plus items relating to provisions for recovery from the destruction of data processing capabilities (hardware, software, or data). Items relating to efficiency and effectiveness are not included.

There are a variety of methods for organizing and wording the questions in an internal control review questionnaire. The method used in this illustrative questionnaire is to have a question to summarize the auditor's overall assessment of the controls relating to a control objective. For example, "Is there suitable segregation of functions between the EDP department and users?" In addition to a YES or a NO response, the auditor can describe the basis for the response. The summary question is followed by a list of control features that are frequently used to achieve the control objective. The summary question and the control features list are worded so that a YES response indicates the control is said to be present or is adequate and a NO response indicates the control is absent or inadequate. Not all control features need be present to achieve adequate control, the list is used as an aid in formulat-

ing the summary response and to provide factual documentation. Comments may be added and reference may be made to accompanying schedules, tests of compliance, and so forth. There is no description of the control features listed, since they are explained in the book.

## Illustrative Internal Control Review Questionnaire Computer Data Processing

### Part 1: Background Information

Prepared by \_\_\_\_\_ Date \_\_\_\_\_

Reviewed by \_\_\_\_\_ Date \_\_\_\_\_

Client Organization \_\_\_\_\_

Location of EDP \_\_\_\_\_  
facilities being reviewed \_\_\_\_\_

Name of individual in charge at this location \_\_\_\_\_

Telephone ( ) \_\_\_\_\_

### 1.1 Hardware Configuration

1 1 1 Attach schedule describing overall structure of computer data processing centers (facilities) for this organization. Describe locations and relationships among locations. Identify minicomputers and microcomputers used for data processing. Attach diagram if useful. The following are examples of information to attach for each center:

- Location of data processing center (facility)
- Centralized or decentralized center
- Major purpose(s) of center
- Other centers to which center is connected via data communications and purposes of each. Also identify files that are downloaded from or uploaded to another center.

1 1 2 Attach schedule listing computer data processing equipment at this location. Indicate manufacturer, model number, and quantity for each.

### 1.2 Computer Facilities Layout

1 2 1 Attach description or diagram of computer facilities with respect to physical access.

1 2 2 Attach description of storage area for negotiable document forms (such as checks, stock certificates, etc.).

1 2 3 Attach description or diagram of area for data file (tapes and disks) and program file storage (tapes, disks, documents, etc.) noting physical access characteristics such as doors, windows, pass-throughs, locks, etc.

- 1 2 4 Attach description or diagram of fire extinguishers, computer power shut-off switches, and air conditioning shut-off switches for computer room. Note type of fire extinguishers (water, carbon dioxide or Halon)
- 1 2 5 Attach description of location(s) used for storage of backup media

**1.3 Organization for Computer Data Processing**

- 1 3 1 Attach organization chart for computer data processing including names of personnel holding supervisory positions
- 1 3 2 Attach summary budget figures for computer data processing function at this location
- 1 3 3 Identify and list persons and titles assigned to following control functions

- (a) Librarian function—maintain custody of tapes and disks containing programs and data and documentation books and records \_\_\_\_\_
- (b) Data processing control function—establishes control over input, checks run-to-run controls, controls distribution of output, and resolves exceptions \_\_\_\_\_
- (c) Data backup function—controls preparation and storage of backup copies of files and controls release of backup files when needed for reconstruction or when superseded \_\_\_\_\_
- (d) Access control and security function (if needed)—establishes and monitors access and security for facilities and data \_\_\_\_\_
- (e) Data management administration function (if using database management system) \_\_\_\_\_

- 1 3 4 Attach a description of the internal audit capabilities and identify major computer audit activities performed

**1.4 Computer Software**

- 1 4 1 Identify and list major system software, such as the following
  - (a) Operating system
  - (b) Communications monitor (TP monitor)
  - (c) File management system or database management system
  - (d) Program library system
  - (e) Data dictionary system
  - (f) Interactive programming system
  - (g) Utility programs having audit significance (e.g., programs to modify other programs)
  - (h) Spooling/scheduling system
  - (i) Tape management system
  - (j) Job accounting system
- 1 4 2 Identify and list major application software relevant to financial statements. Note whether application software was acquired from a software vendor or developed in-house. If acquired, identify package name and source. Specify the programming language for the application and the date implemented

<i>Application</i>	<i>X if Auto- mated</i>	<i>Source of Applications In-house Package</i>	<i>If Package, Name</i>	<i>Program Language and Date Implemented</i>
<hr/>				
(a) Revenue Cycle				
(1) Processing customer orders				
(2) Granting credit				
(3) Delivering goods and services				
(4) Billing customers and recording sales				
(5) Collecting and recording cash receipts				
(6) Maintaining accounts receivable records				
(7) Costing sales				
(8) Maintaining warranty records				
(9) Processing and recording sales commissions, returns and allowances, sales taxes, etc				
(10) Other				
<hr/>				
(b) Expenditure Cycle—				
Purchasing/Disbursements				
(1) Vendor selection and vendor records				
(2) Requisition processing				
(3) Purchasing				
(4) Receiving and recording goods and services purchased				
(5) Maintaining accounts payable records				
(6) Processing and recording cash disbursements				
(7) Other				
<hr/>				
(c) Expenditure Cycle—				
Payroll				
(1) Personnel selection and personal records				
(2) Timekeeping/records				
(3) Payroll accounting, employee benefits, and withholdings				
(4) Payroll disbursements				
(5) Other				
<hr/>				

<i>Application</i>	<i>X if Auto-mated</i>	<i>Source of Applications In-house Package</i>	<i>If Package Name</i>	<i>Program Language and Date Implemented</i>
<hr/>				
(d) Production/Conversion Cycle				
(1) Production order processing/scheduling				
(2) Material requirements planning				
(3) Production cost accounting records				
(4) Property, plant and equipment and depreciation				
(5) Other				
<hr/>				
(e) Financing/Investing				
(1) Banking records and bank reconciliation				
(2) Cash flow analysis				
(3) Securities/investment portfolio records				
(4) Short/long term debt records				
(5) Interest expense/income and dividends				
(6) Shareholder records				
(7) Other				
<hr/>				
(f) Financial Reporting Cycle				
(1) Journal entry processing				
(2) General ledger posting				
(3) Other				

(cont.)

**Part 2: General Controls for Computer Data Processing**

Prepared by \_\_\_\_\_ Date \_\_\_\_\_

Reviewed by \_\_\_\_\_ Date \_\_\_\_\_

Name of organization \_\_\_\_\_

Data processing center or facility being reviewed \_\_\_\_\_

**2.1 Organization and Operation Controls**

2 1 1 Is there suitable segregation of functions between EDP and users served by it? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain:

<i>In making the assessment, consider the following.</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) Policy that EDP department does not originate or authorize transactions EDP department does not correct errors originating with users			
(b) EDP department does not have custody or control over non-EDP assets			
(c) EDP department does not have authority to initiate master file changes			
(d) Standard procedure for post processing review by user area of all transactions automatically generated during computer processing			
(e) Standard procedure for post processing review by user area of all master file changes			
(f) Accounting maintains independent control totals for general ledger			
(g) Custody and control over negotiable paper stock (such as blank check stock) includes non-EDP personnel			



2 1 2 Are there organizational units external to EDP which perform suitable review and control activities of the EDP function? Explain Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) EDP steering committee or users committee			
(b) Annual budget and performance review by top management			
(c) Regular internal audit of EDP function			

2 1 3 Is there suitable segregation of functions within the EDP department? Explain Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) Separation of programming and systems analysis from operations			
(b) Separate processing control function to control inputs, outputs, and distribution of outputs and resolve exceptions			
(c) Separate librarian function with custody of files, programs and documentation			
(d) Separate data administration function (if database management software is used)			
(e) Separate access control and security function (if needed in this center)			
(f) Regular rotation of job run duties by operators			
(g) Rotation of application program maintenance duties			
(h) Vacations are taken regularly by employees of data processing			

(cont.)

**2.2 Application System Development and Maintenance Controls and Documentation**

2.2.1 Is there adequate management and user participation and review to ensure approval of application development according to management specifications? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following.</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) New application systems are part of master plan approved by management			
(b) New application system development authorized by users			
(c) Users are involved in specifications for new applications to be acquired or built			

2.2.2 Is there a well-controlled process for development, testing, and conversion of new applications? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) There is a formal procedure for application system development with well-defined tasks, phases, deliverables, approvals, etc			
(b) Internal Audit reviews the controls in new systems during the development cycle			
(c) Program development review and sign-off by data processing supervisor and user management			
(d) Life cycle test procedures are comprehensive and complete			
(e) File conversions are well planned in life cycle and well controlled			

2 2 3 Is maintenance (updating and changing) of applications adequately controlled? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

	<i>Control Feature Exists</i>	
<i>In making the assessment, consider the following</i>	<u>Yes</u> <u>No</u>	<i>Comments and/or Reference</i>
(a) All proposed program changes are documented and approved before being made		
(b) Operations personnel are not authorized to make application program corrections		
(c) All program changes are tested and the changes reviewed by independent EDP reviewer and user management		
(d) All program changes and testing are documented		
(e) Library control software is used to control source program versions and object programs		
(f) Use of system software to make changes in object programs and bypass source program change controls is not allowed or is under strict control (example is SUPERZAP)		

2 2 4 Are documentation standards adequate for application maintenance, user instruction, and audit review? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

	<i>Documentation Feature Exists</i>	
<i>In making assessment of documentation standards, consider possible elements of documentation</i>	<u>Yes</u> <u>No</u>	<i>Comments and/or Reference</i>
(a) Application system documentation (1) narrative describing system (2) overall system flowchart (3) description of inputs, files, and outputs (4) list of programs (5) list of manual operations (6) list of controls		

(cont.)

<i>In making assessment of documentation standards, consider possible elements of documentation</i>	<i>Documentation</i>		<i>Comments and/or Reference</i>
	<i>Feature Exists</i>		
	<i>Yes</i>	<i>No</i>	
(b) Run or job documentation (1) narrative (2) run-to-run diagram (3) summary of programs (4) run frequency and estimated timings (5) source documents files and reports for application			
(c) Program documentation (1) narrative (2) flowcharts, hierarchy charts or pseudocode (3) source code listing (4) test data and sample results (5) change notices			
(d) File documentation (1) part of program documentation or (2) separate data dictionary used			
(e) Operations documentation (1) job and program narratives (2) input/output chart for files (3) job command list (4) error message list (5) restart and recovery instructions			
(f) User documentation (1) system description (2) operation and control procedures			

**2.3 Hardware Controls**

*Note* The hardware controls in computer equipment are generally adequate and automatic. In the absence of errors assignable to hardware, the auditor can rely upon hardware controls.

Have there been hardware errors which have affected the processing of transactions or the storing or accessing of data? Yes \_\_\_\_\_ No \_\_\_\_\_

Answer based on inquiry of \_\_\_\_\_

or examination of records \_\_\_\_\_

If yes explain \_\_\_\_\_

**2.4 System Software Controls**

2 4 1 Are system software features useful in control being utilized? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) File labeling and checking provisions			
(b) Program and file access provisions (such as passwords)			
(c) Debugging aids			
(d) Accounting information provisions			
(e) Usage statistics			

2 4 2 Are system software changes subject to adequate controls? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) System software changed only after authorization			
(b) Review and testing of all system software changes by person other than one making the changes			
(c) System software change commands can be entered only from an authorized master terminal			

(cont.)

**2.5 Access Controls**

2 5 1 Is access to documentation restricted to persons who require it for authorized duties? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) Documentation stored in secure location			
(b) Access and use of documentation logged by document custodian			
(c) Out-of-date documentation shredded or otherwise destroyed			

2 5 2 Are the duties of data processing librarian well defined and adequately supervised? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following duties</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) Has custody and controls access to all data files on tape, disk packs, etc			
(b) Monitors and controls access to program libraries			
(c) Has custody and controls access to documentation			
(d) Keeps record of all data file use			
(e) Monitors all program updating and use			
(f) Issues files only for scheduled jobs			
(g) Makes use of library control software to control programs in source and object code and control test versions and production versions			
(h) Takes periodic inventory of program libraries and data files			

2 5 3 Is access to computer facilities adequately restricted? Include alternate entry points such as emergency exit in evaluation Explain Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider possible access control methods</i>	<u>Used</u>		<i>Comments on Effectiveness and Compliance</i>
	Yes	No	
(a) Locked area with keys for authorized personnel			
(b) Receptionist or guard			
(c) Card-key for admission			
(d) Code device			
(e) Other _____			

2 5 4 Is access through terminals to computer processing and data files adequately safeguarded? Explain Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider methods for controlling access through terminals</i>	<u>Used</u>		<i>Comments on Effectiveness and Compliance</i>
	Yes	No	
<b>Physical Controls</b>			
(a) Terminal(s) in separate locked room			
(b) Terminals have keylock for on/off switch			
(c) Certain transactions restricted to specified terminals			
<b>Software Controls</b>			
What software enforces access procedures _____			
(d) Personal security access code required for each terminal session			
(e) Personal security access code required for each sensitive transaction			
(f) Access violations cause terminal to be disconnected			
(g) Disconnected terminal requires supervisor approval to reconnect			
(h) Certain transactions restricted to specified hours of operations			
(i) Access security code required to access sensitive data elements			

326 Appendix A

2 5 5 Are security codes for terminal access adequately controlled? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

Are passwords assigned to each individual or to groups? \_\_\_\_\_

<i>In making the assessment, consider methods for controlling security codes</i>	<u>Used</u>		<i>Comments on Effectiveness and Compliance</i>
	Yes	No	
(a) Security code changed frequently (say weekly)			
(b) Security code assigned as set of random characters			
(c) Enforced policy that security code is not written down by users			
(d) Security code not printed out or displayed by terminal			
(e) Security code table in computer protected from unauthorized access			

2 5 6 Is there adequate screening and instruction of new employees before access to computer facilities is granted? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following new employee access procedures</i>	<u>Exists</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Background check on all employees			
(b) Clear, enforced policy on employee use of equipment, unauthorized program changes, unauthorized access, unauthorized or careless issuance of security codes, keys, etc			

2 5 7 Is access to computer facilities restricted for terminated employees? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following for terminated employees</i>	<u>Done</u>		<i>Comments on Effectiveness and Compliance</i>
	Yes	No	
(a) Immediate notification of all personnel controlling access to facilities, library, backup file, storage, etc			
(b) Immediate return of keys and cards used for access to facilities			



<i>In making the assessment, consider the following for terminated employees</i>	<u>Done</u>		<i>Comments on Effectiveness and Compliance</i>
	Yes	No	
(c) Immediate severance or transfer to work area without access to computer facilities			
(d) Immediate removal of employee s terminal access code from access table in computer			

2 5 8 Is there adequate, regular, supervisor review of access and use activity and prompt review and investigation of violations of procedures for access and use? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider possible areas needing access and use review</i>	<u>Reviewed</u>		<i>Comments on Effectiveness and Compliance</i>
	Yes	No	
(a) Review of librarian function control of access and use of documentation and files			
(b) Review of terminal use for monetary transactions and master file changes			
(c) Review of procedures for investigation of terminal access violations			
(d) Review of system use data (system log or system accounting facility)			

**2.6 Operation Controls**

2 6 1 Is there a separate data processing control function associated with EDP? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

2 6 2 Are the duties of the data processing control function (control clerk or control group) well defined and adequately supervised? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment consider possible duties</i>	<u>Assigned</u>		<i>Comment and/or References</i>
	Yes	No	
(a) Logging of input data			
(b) Recording of control information			
(c) Reconciliation of control information			
(d) Recording of progress of job through EDP			

(cont.)

<i>In making the assessment, consider possible duties</i>	<u>Assigned</u>		<i>Comment and/or References</i>
	Yes	No	
(e) Supervision of output distribution			
(f) Scrutiny of console logs and printed control information			
(g) Liaison with users regarding errors			
(h) Logging of corrections			
(i) Scrutiny of error listings			
(j) Maintenance of error log or error report			

2 6 3 Is there a well-controlled and well-defined general procedure for logging erroneous transactions, distributing notices to persons responsible for corrections, logging corrections, and reentering corrected data? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider possible techniques</i>	<u>Used</u>		<i>Comment and/or Reference</i>
	Yes	No	
(a) Manual error log			
(b) Manual procedure for error notification, followup, etc			
(c) Automated error log such as circulating error file			

2 6 4 Are operations performed in a controlled, orderly manner so as to provide reasonable assurance of accuracy and completeness? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>Examples of supervision and control for operations</i>	<u>Used</u>		<i>Comment and/or Reference</i>
	Yes	No	
(a) Adequate supervision of operations			
(b) Adequate operator instructions			
(c) Orderly, planned scheduling of jobs			
(d) Orderly work flow			
(e) Good housekeeping practices			
(f) External labels on all tapes and disk packs External labels should normally contain a reference to a directory rather than description of contents			

2 6 5 Is there a review on a continuing basis of computer processing operations by internal auditors and other independent review group? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain:

**2.7 Database Controls**

2 7 1 Is there a separate database administration function within EDP? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

2 7 2 Are the duties of the database administration function well defined and adequately supervised? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider possible duties</i>	<u>Assigned</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Control over database software			
(b) Establish and control database definition			
(c) Establish database access control procedures, monitor compliance, and review violations			
(d) Establish and maintain database backup and recovery procedures			
(e) Maintain database documentation			
(f) Monitor performance of database operations			
(g) Perform database quality testing procedures			

2 7 3 Are establishment and changing of databases maintained by database management systems adequately controlled? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following</i>	<u>Control Feature Exists</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Access to database change utilities controlled and use logged and reviewed			
(b) All maintenance of databases logged and reviewed by database administrator			

(cont.)

330 Appendix A

<i>In making the assessment, consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(c) Enforced standards for data definition			
(d) Enforced standards for database changes			
(g) Database usage reporting procedures with investigation of unusual activity			

2 7 4 Is there adequate up-to-date documentation of data items in the files and databases? Yes \_\_\_\_\_ No \_\_\_\_\_  
Explain

<i>In making the assessment, consider the following</i>	<i>Control Feature Exists</i>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) Automated data dictionary			
(b) Manual data dictionary			
(c) Data divisions associated with each file			

**Part 3: Continuation Provisions**

Prepared by \_\_\_\_\_ Date \_\_\_\_\_

Reviewed by \_\_\_\_\_ Date \_\_\_\_\_

Name of organization \_\_\_\_\_

Location being reviewed \_\_\_\_\_

**3.1 Insurance and Risk Review**

3 1 1 Is there a periodic (say annual) review of risks to continued operations and followup to remove reduce or insure against exposures? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain \_\_\_\_\_

<i>Risks which should be considered for review</i>	<i>Reviewed</i>		<i>Comments and/or Reference Dates of Last Review</i>
	<i>Yes</i>	<i>No</i>	
(a) Fire			
(b) Flood			
(c) Wind (tornado)			
(d) Earthquake (if applicable)			
(e) Riot or civil commotion			
(f) Power interruption or power fluctuations			
(g) Unauthorized access and use			
(h) Strikes by employees including data processing employees			

3 1 2 Is there adequate insurance to cover costs of interruption and costs of restoring operations in the event of a disaster? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain \_\_\_\_\_

<i>In making the assessment evaluate insurance for the following</i>	<i>Exists</i>		<i>Coverage Limits</i>	<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>		
(a) Replacement of hardware and system software				
(b) Costs of reconstruction of programs and data on tape disk, etc				
(c) Business interruption costs and additional expense of recovery				

**332 Appendix A**

3 1 3 Is there adequate fidelity insurance (bonding) to recover costs of employee fraud or misappropriation, etc ? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

**3.2 Disaster Contingency Plan**

3 2 1 Is there a disaster contingency plan? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, evaluate disaster plan for</i>	<u>Part of Plan</u>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
<i>(a) Alternate power source</i>			
<i>(b) Backup hardware and backup site</i>			
<i>(c) Backup software</i>			
<i>(d) Backup data files</i>			
<i>(e) Backup business forms and other supplies</i>			
<i>(f) Backup documentation for programs, operations, etc</i>			
<i>(g) Staffing of backup operation</i>			

3 2 2 Is the disaster contingency plan reviewed and tested periodically (say, once a year)? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain  
 Date of last test \_\_\_\_\_

3 2 3 Is there a person assigned to be in charge of maintaining the disaster plan? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

**3.3 Offsite Storage and Backup Storage**

3 3 1 Is offsite storage reasonably well protected? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following features for offsite storage</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
<i>(a) Protection against unauthorized access</i>			
<i>(b) Protection against fire</i>			
<i>(c) Protection against water damage from sprinklers, broken pipes, etc</i>			
<i>(d) Not vulnerable to riot or commotion</i>			
<i>(e) Environmental protection from heat humidity, etc</i>			

3 3 2 Is offsite backup storage being adequately used? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following backup provisions</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	<i>Yes</i>	<i>No</i>	
(a) Storage of tape and disk packs containing grandfather and greatgrandfather files and related transactions Also periodic database copies and transaction logs			
(b) Storage of backup tapes or disk packs containing program library and system software			
(c) Storage of backup documentation			
(d) Storage of backup supply of forms			
(e) Storage of disaster plan			

3 3 3 Is backup site well protected and/or at sufficiently different location as to not be at risk from the same event as the main site? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain.

3 3 4 Has reconstruction been performed using programs, data, etc , from backup location? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

**Part 4: Audit Questionnaire for Evaluation of Application Controls**

*One for each application included in audit*

---

Application \_\_\_\_\_

---

---

---

Prepared by \_\_\_\_\_ Date \_\_\_\_\_

Reviewed by \_\_\_\_\_ Date \_\_\_\_\_

Client organization \_\_\_\_\_

---

**4.1 Information About Application**

- 4 1 1 Attach a description of the application describing purpose, document inputs, document outputs (reports, transaction documents, listings), and terminal screen inputs and outputs
- 4 1 2 Describe input and output relationship of this application to other applications
- 4 1 3 Describe general ledger accounts affected by this application
- 4 1 4 Attach description of financial and operational effects of following uncorrected errors if they occur with this application
  - (a) Unprocessed transactions
  - (b) Errors in critical input data item fields (identify the critical data items)
  - (c) Errors in critical data items in application files (identify the critical data items)
  - (d) Errors in critical computations
  - (e) Unauthorized access to application outputs
  - (f) Loss of application outputs
  - (g) Unauthorized access to application files
  - (h) Loss of application files
- 4 1 4 Describe duties and responsibilities of users of application with respect to
  - (a) Input documentation preparation
  - (b) Entry using terminal
  - (c) Preparing correction and adjusting transactions
  - (d) Use of output
  - (e) Initiating, authorizing, and reviewing master file changes or database changes
  - (f) Performing control activities such as maintaining and checking control totals testing output for correctness etc



**4.2 Initiation of Transactions**

4 2 1 Are there adequate controls to ensure that all necessary transactions are initiated on a timely basis?

		Control Adequate
<u>List of Transactions</u>	<u>Initiation Method Used (See Below)</u>	<u>Yes</u> <u>No</u>

Examples of initiation techniques and related controls

- (a) Transaction initiated as part of or byproduct of recording interaction with customers, suppliers, and other units in organization. There are well-defined procedures and responsibilities for initiating the transaction and accounting for records.

---

- (b) Transaction initiated from query, request, or informal order document received. Log of documents received is kept and initiation of transaction is recorded. Log review identifies transactions not initiated. (The log can also be used for telephone queries or verbal requests.)

---

- (c) Transaction initiated by formal order or action document from another unit, supplier, etc., which keeps copy of document to match with copy of transaction document initiated. If transaction is not initiated, the original source of the transaction will produce an error query. There are independent units within the organization charged with the responsibility for following-up on all such error queries.

---

- (d) Transaction automatically initiated by this application or another application as result of another related, concurrent or prior period transaction. Adequate control information and audit trails are maintained to ensure that all necessary transactions were generated and to ensure that none were duplicated (anticipation control).

---

- (e) Transaction initiation prompted by application from schedule of transactions needed. There is periodic checking of transactions against the schedule to ensure all transactions prompted were initiated (anticipation control).

---

- (f) Transaction prompted by application in order to complete other processing. There is periodic checking of transactions against the application output to ensure that all transactions prompted were initiated.

---

- (g) Transaction prompted by error file record. There is appropriate review of error suspense files to ensure prompted transactions were initiated.

---

**4.3 Data Preparation Using Batched Documents**

4 3 1 Are there adequate controls over recording of authorized data on documents and preparation of input batches?

Yes \_\_\_\_\_ No \_\_\_\_\_

Explain

<u>Examples of techniques that may be appropriate</u>	<u>Used</u>	<u>Comments and/or Reference</u>
	<u>Yes</u> <u>No</u>	
(a) Forms with labels, boxes, etc., to aid correct and complete recording and authorization		

(cont.)

<i>Examples of techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(b) Turnaround documents prepared by prior processing and submitted as documents for processing			
(c) Review of documents before submission for processing			
(d) Preparation of one or more control totals for batch of documents being submitted. Examples are (1) Count of records (2) Sum of financial items (3) Hash totals of non-financial items			
(e) Pre-numbered forms, if feasible, and sequence checking by computer to account for all documents			
(f) Authorization initials or signature on batch and/or individual documents (or authorization is otherwise documented)			
(g) Data batch logged by user area and sent to data processing with transmittal form containing control totals			

4 3 2 Is the audit trail at data preparation adequate to trace data to batch or back to individual source documents? Explain Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider the following</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Batch is assigned batch number and documents stored by batch number			
(b) Processing uses batch number as identification			
(c) Processing uses document numbers and documents stored by document number			
(d) Processing uses dates and documents stored by processing period			
(e) Adequate document retention policies			

4 3 3 Are there adequate controls over conversion of data to machine-readable form? Explain Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider the following</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Data batches received are logged by data preparation function			
(b) Verification of data conversion (1) Second key entry for verification comparison (2) Visual verification (3) Check digit verification (4) Programmed check for valid data			
(c) When data batches have been converted to a machine-readable form, the batches of documents are cancelled and logged out			
(d) Control totals are checked at data conversion and batch is balanced			

**4.4 Controls Over Direct Entry at Terminal Without Batching of Documents**

4 4 1 Is there adequate control over authorization of transactions being entered at terminal? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain.

<i>In making the assessment, consider the following terminal entry controls:</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Authorization on documents used in entry process			
(b) Transactions authorized by standard procedure			
(c) Transaction authorized by access privilege (1) Restricted to specific terminals (2) Restricted to specific access passwords assigned to designated individuals			
(d) Authorization at subsequent review of listing of transactions as grouped into logical batches			

Software used to enforce terminal entry controls

(cont.)

338 Appendix A

4 4 2 Are controls adequate to assure complete, correct, timely entry of data at terminal?  
Explain

Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider the following terminal entry controls</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Entry terminal keyboard designed to aid correct input (special keytops, templates, function keys, etc )			
(b) Video terminal screen layout designed to aid correct and complete input			
(c) Echo of input data plus descriptive data for visual validation of data			
(d) Check digit on codes checked at entry			
(e) Comparison of codes at entry with stored table of values to check for valid classification, valid transaction code, etc			
(f) Range and reasonableness tests for numeric input items being entered			
(g) Valid item size, sign, and composition tested for input data items			
(h) Test for duplicate input			
(i) If entry is interrupted, restart procedures control for loss of entered data or duplication of data			
(j) Log of all input transactions for backup and logical batch lists			
(k) Subsequent user or other review of listing of transactions as grouped into logical batches			
(l) Sequence checking to account for all documents (if input documents pre-numbered and controlled)			

4 4 3 Is there an adequate audit trail created at terminal data entry?  
Explain

Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider the following controls that may be appropriate</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Input transactions are assigned a transaction identifier by entry terminal Identifier may include time, terminal number, user I D , etc			

<i>In making the assessment consider the following controls that may be appropriate</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(b) Document identifiers are entered as input for documents (if any) used in data entry			
(c) Documents (if any) used in entry are marked and stored so they can be traced			
(d) Listing of transactions by logical batch is prepared (with adequate frequency) Examples of logical batch listings are transactions by transaction number, by terminal by operator, by account, and by application			
(e) Documents and/or listings contain sufficient reference to trace to authorization for transaction			

4 4 4 Is there adequate control on overrides of processing controls and supervisor adjustments? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider the following controls that may be appropriate</i>	<u>Used</u>		<i>Comments and/or Reference</i>
	Yes	No	
(a) Log of all terminal operator overrides is prepared and reviewed by supervisors			
(b) Overrides restricted to supervisors with special keys or override codes			
(c) Supervisor overrides and adjustments are logged and reviewed by independent reviewer			

**4.5 Input Data Validation and Error Handling**

4 5 1 Is data validation handled by a generalized validation module or by individual program module? \_\_\_\_\_

4 5 2 Is data validation adequate? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

(cont.)

340 Appendix A

<i>In making the assessment, consider the validation techniques that may be appropriate</i>	<u>Used</u>		<i>If Used, Which Data Items Checked</i>	<i>Comments and/or References</i>
	Yes	No		
(a) Logic tests				
(1) Valid classification or transaction code				
(2) Valid item size, sign, and composition				
(3) Invalid values				
(4) Valid combination				
(5) Missing data test				
(6) Sequence test				
(7) Range and reasonableness test				
(b) Check digit				
(c) Stored data comparison				
(d) Control total comparison				
(e) Anticipation control				

4 5 3 Are procedures for error notification and resubmission adequate to prevent loss of erroneous input records returned for correction and prevent duplicate corrections?  
Explain

Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider possible control techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Error record returned to originator who has full responsibility for correction and reentry			
(b) Input held in automated error file until corrected and reentered (circulating error file)			
(c) Erroneous record stored in application file but marked with error flag until corrected			
(d) Erroneous record placed in application data suspense file until corrected			
(e) Manual or automated error log to record errors and receipt of corrections			
(f) Batch register list for information about errors and their disposition			
(g) Reports to user areas of uncorrected errors			

4 5 4 Are there adequate procedures to ensure complete and correct data in error corrections?  
Explain

Yes \_\_\_\_\_ No \_\_\_\_\_

<i>In making the assessment, consider possible control techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Sufficient error report output to permit users to identify errors and make corrections			
(b) Unambiguous identification of record and fields being corrected			
(c) Corrected input subject to complete validation			
(d) Use of error report as control document for recording correction or other disposition			
(e) Explicit instructions and responsibilities assigned for error correction			

**4.6 Processing Controls**

4 6 1 Are controls adequate to ensure processing is complete and accurate? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider techniques that may be appropriate</i>	<u>Used</u>		<i>Explanation Comment and/or References</i>
	Yes	No	
(a) Checking of internal file labels to validate correct version of file is in use			
(b) Test of input record to assure correct record being processed			
(c) Printout of values obtained from internal table and used in computation			
(d) Printout to identify default value (when default values used)			
(e) Trace data for review of all operator input required to run application			
(f) Crossfooting to detect precision, rounding, and other differences			
(g) Processing logic tests (1) Range and reasonableness (2) Sequence (3) Explicit class identification			
(h) Control figure comparisons			
(i) Error correction screening			
(j) Adequate breakpoints for restart if processing interrupted			

4 6 2 Is data for tracing of processing adequate for control review? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

(cont.)

<i>In making the assessment, consider the following techniques:</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Programmed run-to-run balancing			
(b) Control output showing control totals and identifying out-of-balance conditions			
(c) Control list of all additions, deletions, and changes to internal tables and records in master file giving before-image, change, and after-image			

4.6.3 If blank negotiable forms are used in this application, are there adequate controls over the forms? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain:

<i>In making the assessment, consider the following:</i>	<u>Used</u>		<i>Comment and/or References</i>
	Yes	No	
(a) Negotiable forms are in locked area			
(b) Custody of forms is independent of computer operators			
(c) All forms are accounted for including forms used in printer setup			

**4.7 Output and Stored Data Controls**

4.7.1 Are there adequate controls to ensure that outputs are complete and accurate? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain:

<i>In making the assessment, consider techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Output clearly labelled with date and other identification			
(b) Start and finish of report are clearly specified on output			
(c) Forms or pages are numbered			
(d) Output totals reconciled to input control totals			
(e) Review of reasonableness and completeness by data processing control function			
(f) Review by user department for reasonableness and completeness			
(g) User comparison of report totals and detail items processed with user-maintained controls			



4 7 2 Are controls adequate to ensure distribution of all outputs only as authorized? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Authorized report distribution list for data processing			
(b) Report distribution log by data processing			
(c) Route slips on outputs			
(d) Recipient acknowledgment procedures for classified reports and negotiable documents			
(e) User-area report distribution list and log to record receipt of outputs			
(f) Use of external labels for all files from the application			

4 7 3 Are there regular procedures to ensure that the master files and tables for this application (in separate file or in database) are correct and most current version? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain.

<i>In making the assessment, consider techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Review of list of all master file changes			
(b) Control totals on all files			
(c) Checking of file balancing (old balance plus changes equals new balance)			
(d) Check of master file totals against control figures maintained independently			
(e) Periodic quality review of master file records—footing, cross-footing, and testing of items in the records			
(f) Periodic listing and review of files, in sections or entire file			
(g) Periodic testing of file or sample of file records against physical counts and other data			

(cont.)

**4.8 Continuation Provisions**

4 8 1 Are backup procedures adequate to assure continuation of processing for this application? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Backup copy of programs for this application stored offsite			
(b) Forms and other media needed for processing this application stored offsite			
(c) Operator instructions for this operation stored offsite			
(d) Backup copies of master files and transaction files for this application stored offsite			
(e) User documentation for this application stored offsite			

4 8 2 Are there adequate user procedures for processing vital transactions for this application in the event the computer system is not available? Yes \_\_\_\_\_ No \_\_\_\_\_  
 Explain

<i>In making the assessment, consider techniques that may be appropriate</i>	<u>Used</u>		<i>Comments and/or References</i>
	Yes	No	
(a) Documented procedures for handling terminal transactions in process when central computer system goes down			
(b) Documented procedure to recover from a few minutes of terminal downtime so that no transactions are lost, altered, or duplicated			
(c) Documented procedure for manual data processing for this application			
(d) Documented user procedures for recovery and entry of manual transactions when computer application is restored to service			

---

# Flowcharting

# B

Flowcharts are diagrams of operation sequence, data flow, and processing logic in information processing that are used to describe the proposed system design of data processing applications and to document the completed application. It is important for the auditor to understand flowcharts because they represent significant portions of data processing documentation. The auditor will also find flowcharts useful as concise tools for documenting applications being investigated.

Standard flowchart symbols have been adopted by the American National Standards Institute (ANSI),<sup>1</sup> and the symbols have also been adopted by the International Standards Organization (ISO). Although their use is voluntary, these symbols have been widely adopted, and data processing personnel are encouraged to use them. It is convenient to distinguish between two types of flowcharts even though the same set of symbols is used for both.

- 1 *System flowchart* This chart describes the sequence of major processing operations (both manual and computer) and the data flow to and from the files used in processing.
- 2 *Program flowchart* This chart describes the sequence of operations and logic in a computer program. It is sometimes referred to as a block or logic diagram.

Flowcharts are normally prepared using a template, which is available from computer manufacturers and from independent suppliers. A template is not necessary for preparing charts, but it does add to the speed and neatness with which a good flowchart can be drawn. Special forms are also available to aid in drawing flowcharts, the forms contain predrawn areas in which symbols can be neatly drawn.

As explained in chapter 5, various software packages can prepare program flowcharts from a source program code (computer instructions written in the COBOL, FORTRAN, or assembly languages) and system flowcharts from job control instructions. The flowcharts prepared by this software are usually more detailed than charts drawn by a programmer or an analyst.

## Value of Flowcharts for Documentation

Some practitioners have questioned the value of flowcharts for computer application planning and documentation. The issues in this controversy should interest the auditor because the absence of adequate documentation increases the difficulty of application review in an audit, and the absence of traditional documentation may prompt the auditor to make recommendations to management that he would not make if such documentation were available. The issues are best understood in terms of system flowcharts and program flowcharts.

---

<sup>1</sup> The complete American National Standard X3 5-1970 *Flowchart Symbols and Their Usage in Information Processing* is available from the American National Standards Institute, Inc., 1430 Broadway, New York, New York 10018.

The auditor is generally interested in system flowcharts because they show the overall flow of processing for an application. A system flowchart is a useful and desirable element of documentation for an application. Therefore, it is not unreasonable for the auditor to expect this documentation element for every application. Some installations use alternatives to system flowcharts, such as data flow diagrams, bubble charts, and actigrams, all of which are comparable in scope and purpose to system flowcharts.

Program flowcharts can be used in designing the program logic and in documenting the final result. There are several objections to program flowcharts. Program flowcharts are not used by maintenance programmers who correct or modify existing programs because they do not trust the flowcharts' accuracy. They prefer to work directly with the program instructions. A well-written, structured program with explanatory comment lines imbedded in the instruction code is often sufficient documentation without a program flowchart. A hierarchy chart to show the relationship of program modules is useful, but each program module should be small enough (generally less than fifty lines) and should have sufficient comment lines to be understood by a reviewer. Finally, program flowcharts are time consuming to prepare and difficult to update. On balance, it appears reasonable to replace program flowcharts with hierarchy charts and well-commented program modules. Although there are still cases when program flowcharts are very useful to illustrate the logic and flow of processing, it is not consistent with current directions in EDP practice to require program flowchart documentation for all programs. Also, the auditor does not usually need to review the detailed program logic shown by program flowcharts.

### Summary of Flowchart Symbols

There is one set of six basic symbols for both program flowcharts and system flowcharts. Two of these symbols, the input/output symbol and the process symbol, are general symbols. The input/output symbol specifies any input or output but does not specify the medium or device used. Since it is frequently desirable to indicate the medium or device, specialized symbols for input/output devices or media may be used in place of the general input/output symbol. Likewise, there are specialized process symbols that may be used in place of the general process symbol. Three other symbols, the connector, terminal, and parallel mode, neither process nor input/output symbols, are classified as additional symbols.

The symbols are summarized in figure B-1. The term "on-line" refers to processes and input/output devices connected to, and under the control of, the computer. The term "off-line" refers to processes and devices not connected to, or under the control of, the computer.

**Figure B-1 Basic flowchart symbols**

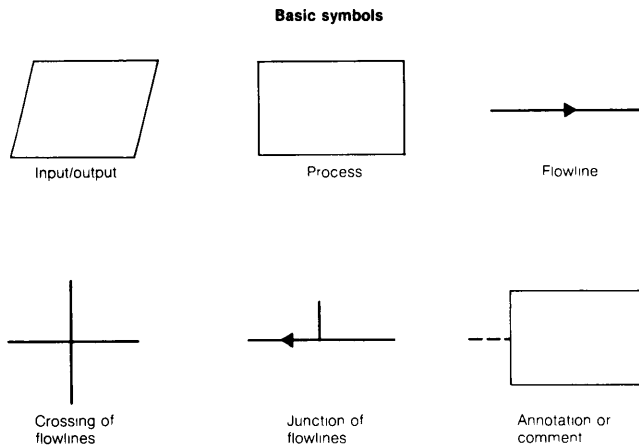
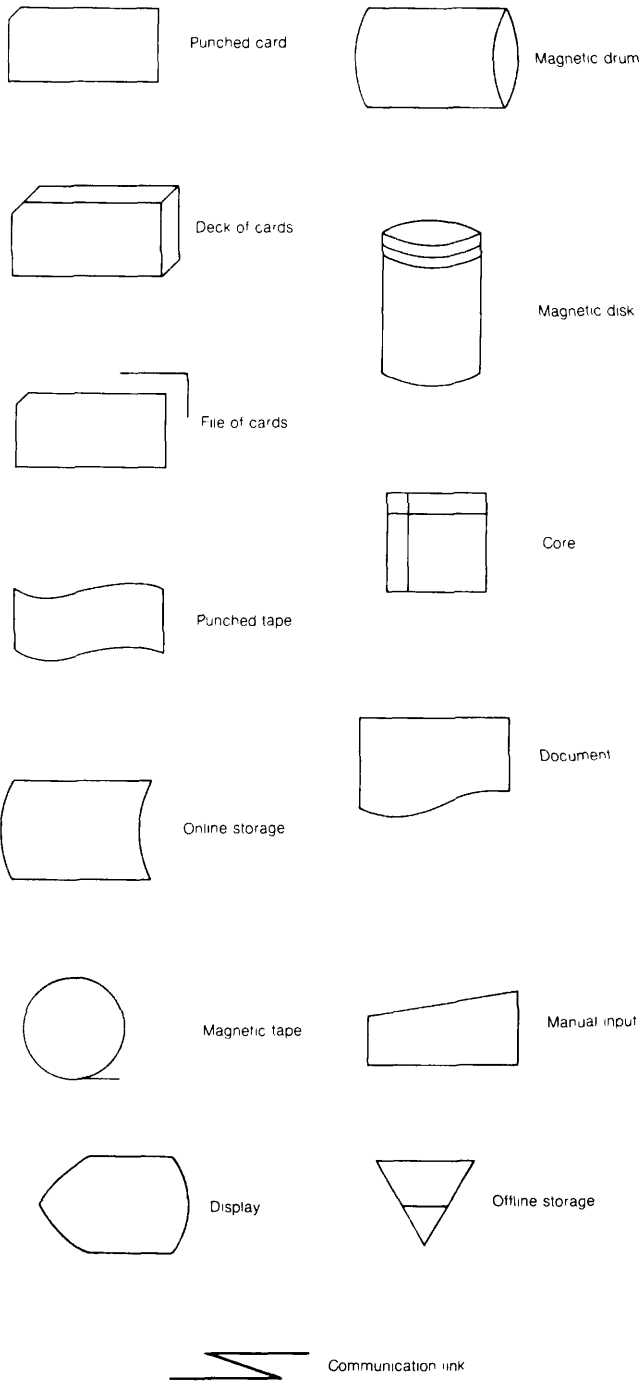


Figure B-1 continued

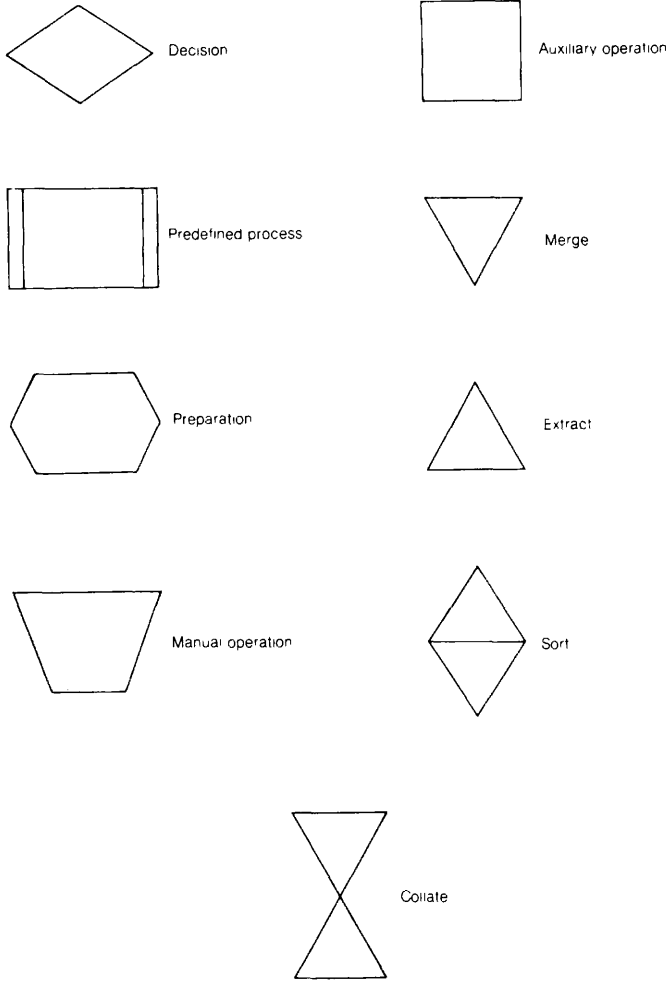
**Specialized input/output symbols**



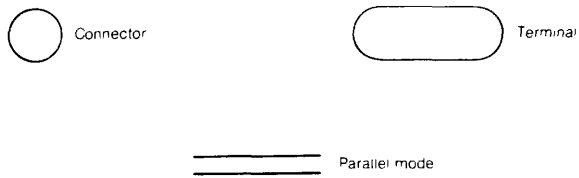
(cont.)

Figure B-1 continued

**Specialized process symbols**

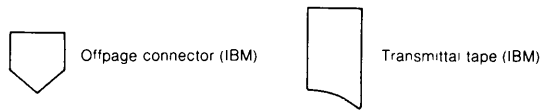


**Other symbols**



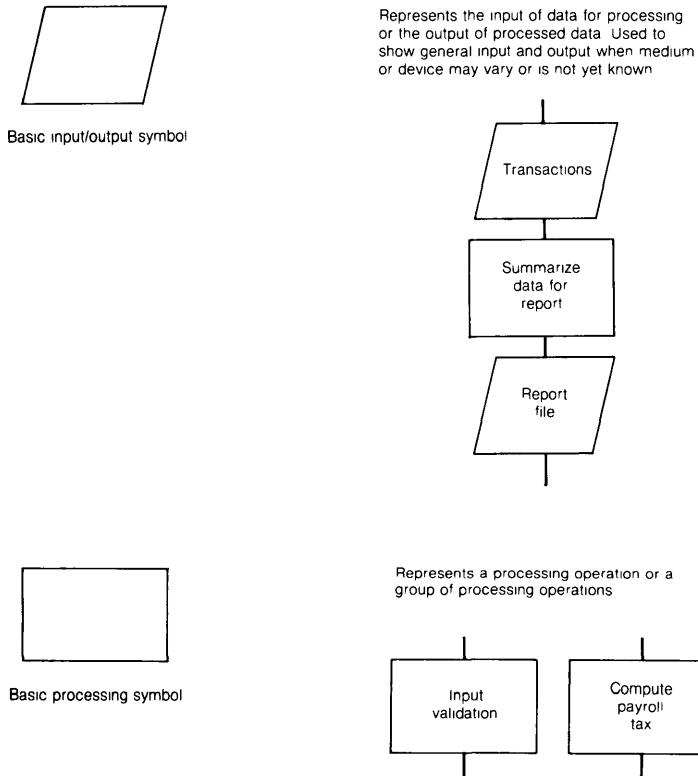
## Use of the Flowchart Symbols

In Figure B-2, the standard symbols are described, and examples of their use are provided. The symbols are connected by flowlines when they are used to prepare flowcharts. The charts are normally read from left to right and from top to bottom. Arrowheads are often included on the flowlines for the sake of clarity, they are required if the flow is in other than the normal direction. A short description of the document, file, process, or decision being represented is usually written inside the symbol. If additional description or explanation is required, an annotation symbol is used.



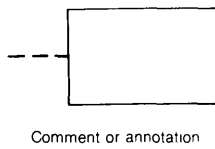
The flowchart symbols presented here are the standard symbols that must be used only in the way they are defined. The flowchart standard does not exclude the use of additional, user-defined symbols. For example, IBM has added symbols for off-page connector and for transmittal tape.

**Figure B-2** Standard flowchart symbols

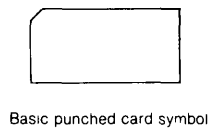
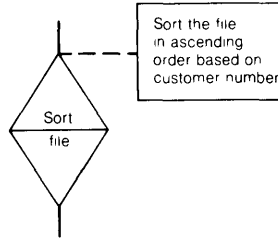


(cont.)

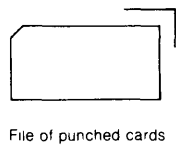
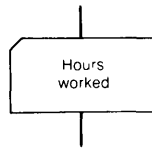
Figure B-2 continued



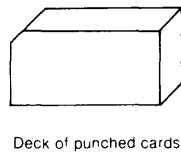
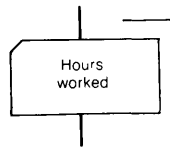
For added comments. Connects to flowchart where meaningful



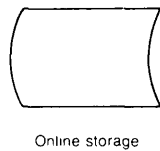
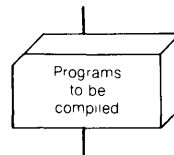
Represents input or output using any type of punched cards



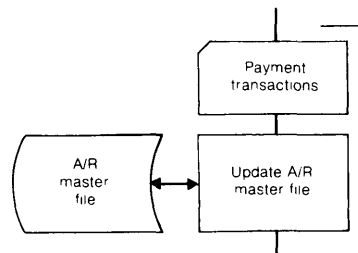
Represents a file of punched cards (i.e. a collection of related punched card records)



Represents a collection of punched cards which may contain unrelated sets of cards



Represents input from or output to any storage that is connected to and under the control of the computer (disk, drum, mass core, magnetic tape, etc.)

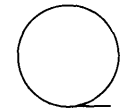
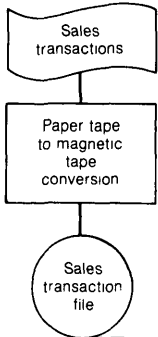






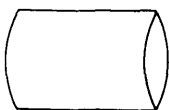
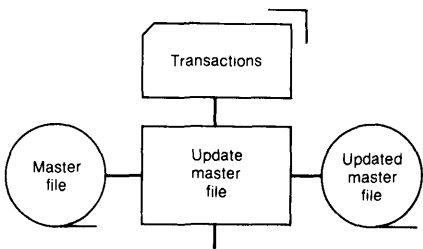
Punched tape symbol

Represents input/output using punched paper tape



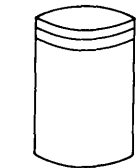
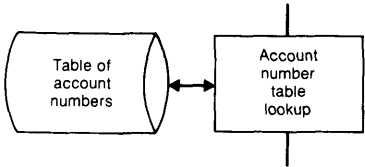
Magnetic tape storage

Represents input/output from magnetic tape storage



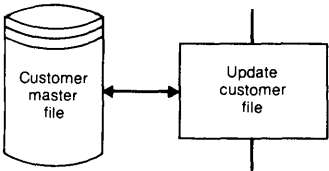
Magnetic drum storage

Represents input/output from online storage using a magnetic drum



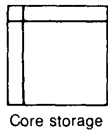
Magnetic disk storage

Represents input/output from either fixed disk storage or disk pack storage

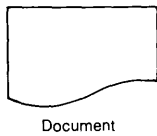
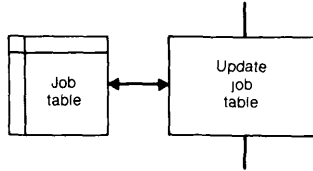


(cont.)

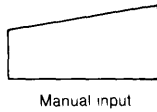
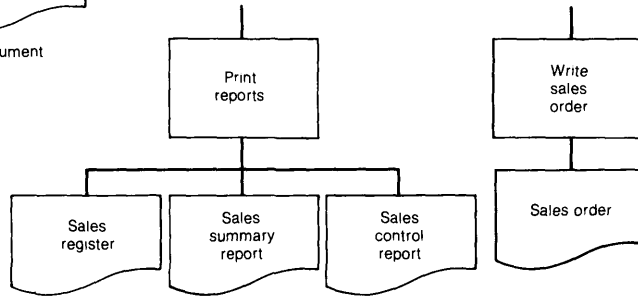
352 Appendix B  
 Figure B-2 continued



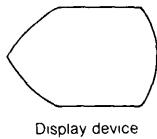
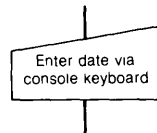
Represents input/output function using magnetic core as storage medium



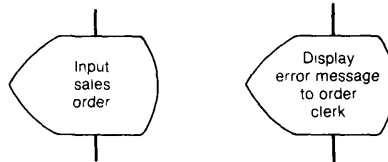
Input/output using any kind of document, e.g. printed output



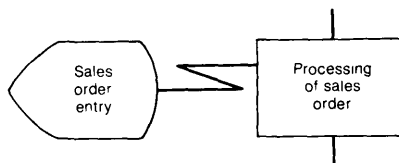
Represents input entered manually at the time of processing e.g. using keyboard, switch settings, etc.

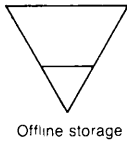


Represents a visual display unit (CRT), typewriter, audio response unit, etc. used during processing for human input or displaying output for human use

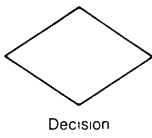
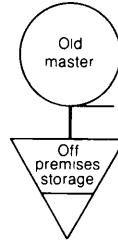
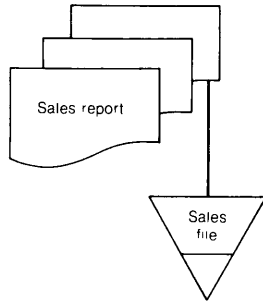


Represents data transmitted by data communications facilities from one location to another. Arrowheads to indicate direction of flow are required if flow is not left to right or top to bottom

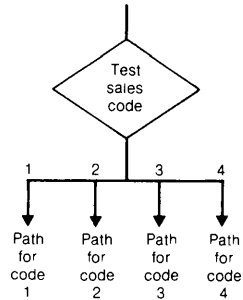
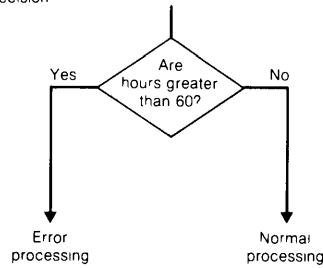




Represents storage not accessible by the computer e.g. paper files, off-premises storage, inactive file storage



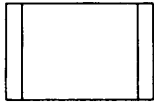
Represents a decision point to determine which of a number of alternative actions of processing paths are to be taken



Notation frequently used in the decision symbol

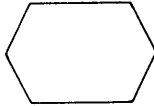
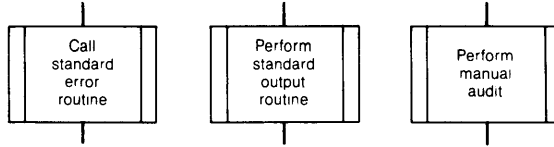
- |   |                          |       |                              |
|---|--------------------------|-------|------------------------------|
| : | compare                  | A : B | compare A with B             |
| > | greater than             | A > B | A greater than B             |
| ≥ | greater than or equal to | A ≥ B | A greater than or equal to B |
| < | less than                | A < B | A less than B                |
| ≤ | less than or equal to    | A ≤ B | A less than or equal to B    |
| ≠ | not equal                | A ≠ B | A not equal to B             |

Figure B-2 continued



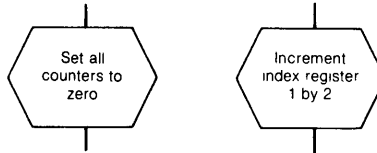
Predefined process

Represents a subroutine or processing module specified in other documentation. Examples are standard modules or subroutines.



Preparation

In program flowchart represents modifications of instruction(s) which change the program. Examples are initializing a routine and program modification procedures such as index register modification.



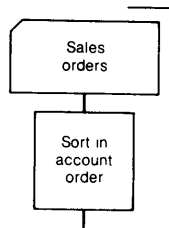
Manual operation

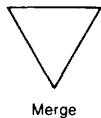
Represents an offline process geared to the speed of a human being. e.g. manual writing of document, manual keying operations, etc.



Auxiliary operation

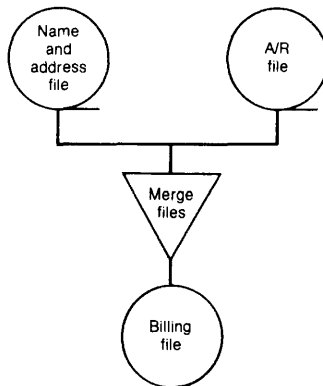
Represents an offline operation using equipment not controlled by the computer. Examples are card sorting using offline sorter, tape-to-tape conversion using offline device, etc.





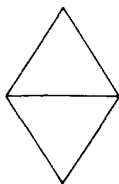
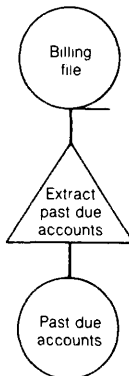
Merge

Represents the merging of two or more sets of items (such as files) into one set (file)



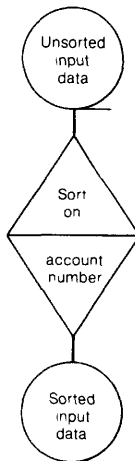
Extract

Represents extraction of items from a set (file)



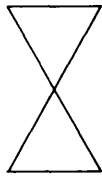
Sort

Represents the sorting of a file or set of items into a sequence



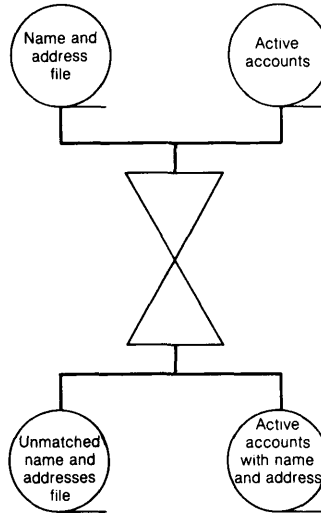
(cont.)

Figure B-2 continued



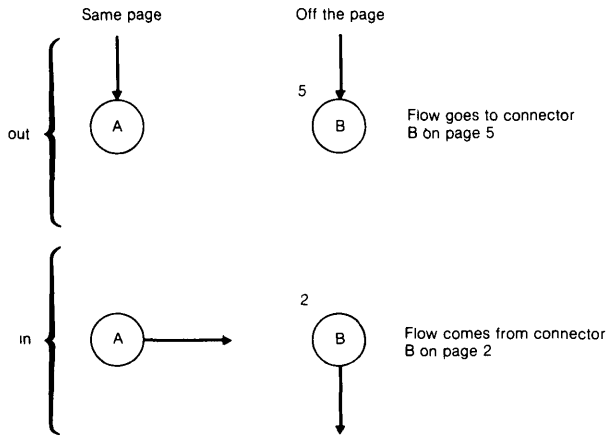
Collate

Represents process of extracting related items from two or more files and merging them to form two or more extracted, merged files



Connector

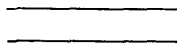
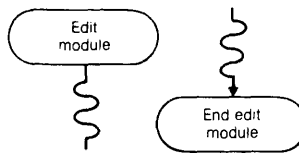
Used to define the flow when limitations such as page size or complexity require a break in the flowline. An outconnector and an inconnector are matched by using a common identifier inside the connector. If the connection is off the page (or otherwise difficult to locate on a large chart), a reference such as page reference is placed at the left of the outconnector and inconnector. A connector is also used at the junction of several flowlines





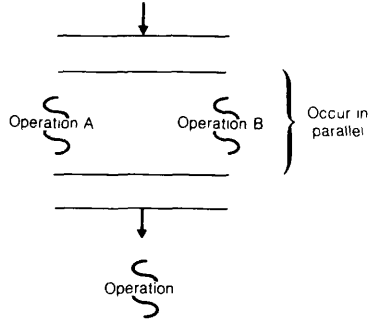
Terminal or Interrupt

In program flowchart represents a starting point or stopping point (halt, delay, stop, or interrupt) in a flowchart



Parallel mode

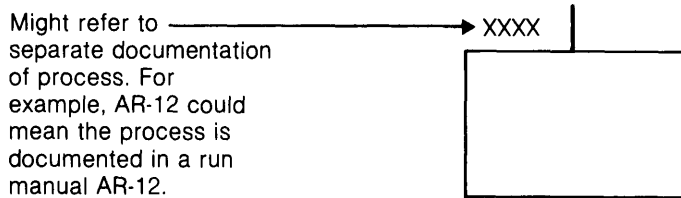
Represents the beginning or end of operations occurring concurrently



## Drawing Flowcharts for Computer Application Systems and Programs

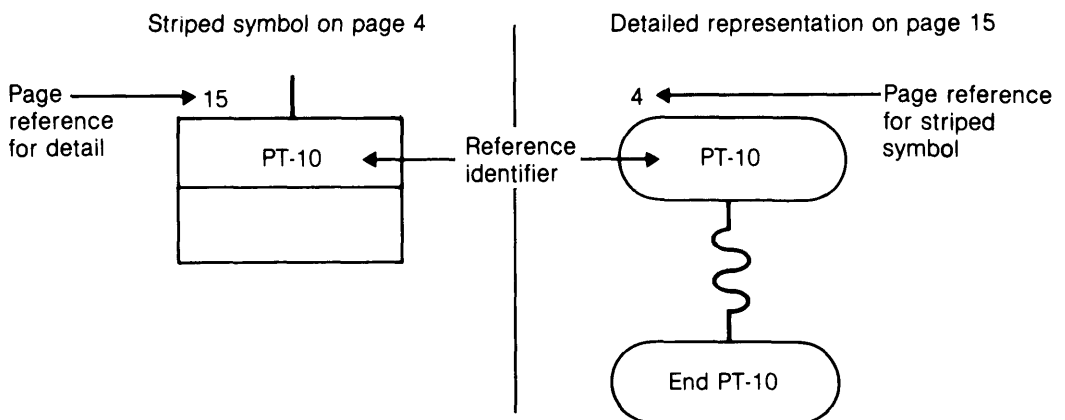
A flowchart reflects, to some extent, the preferences of the person making it. The test of a flowchart is its effectiveness in communicating the logic, operational sequence, and data flow for a computer processing application. The clarity of flowcharts is generally enhanced by standard symbols and by a liberal use of cross-references and other conventions.

Symbols may be cross-referenced to other documentation. The reference is placed above the symbol to the left. International standard reference procedures, which differ somewhat, are not given here.



Symbol striping is used (primarily with the process symbol) to reference a detailed representation of the process in the same set of flowcharts. A horizontal stripe defines an area in the symbol used to reference the detailed program flowchart representation. The detailed representation contains the reference in the terminal symbol that begins the detailed flowchart.

Page references are at the top left of the symbols.





Multiple symbols may be used to specify multiple copies

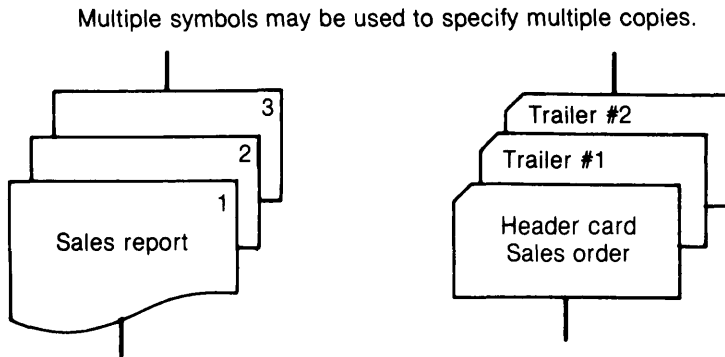
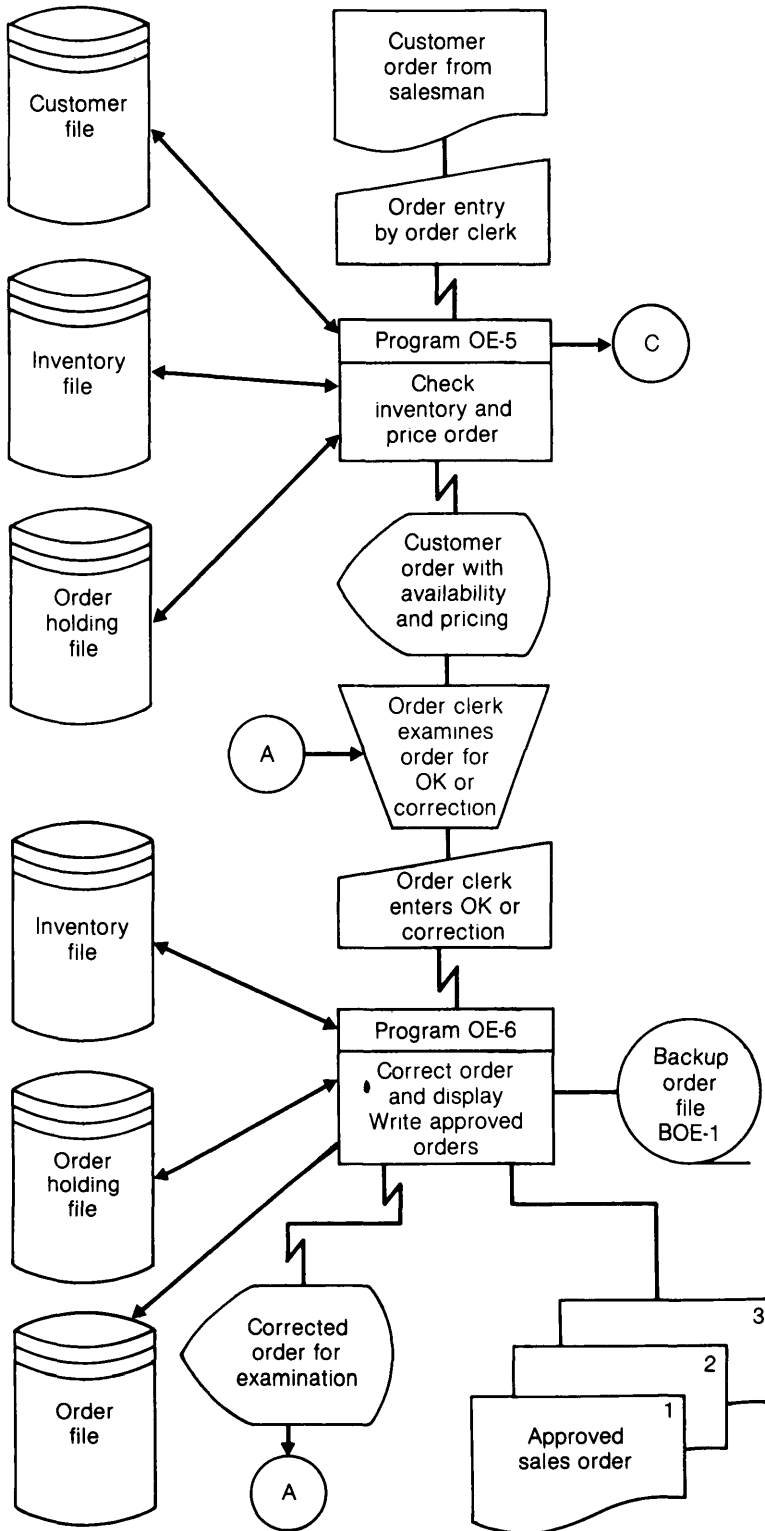
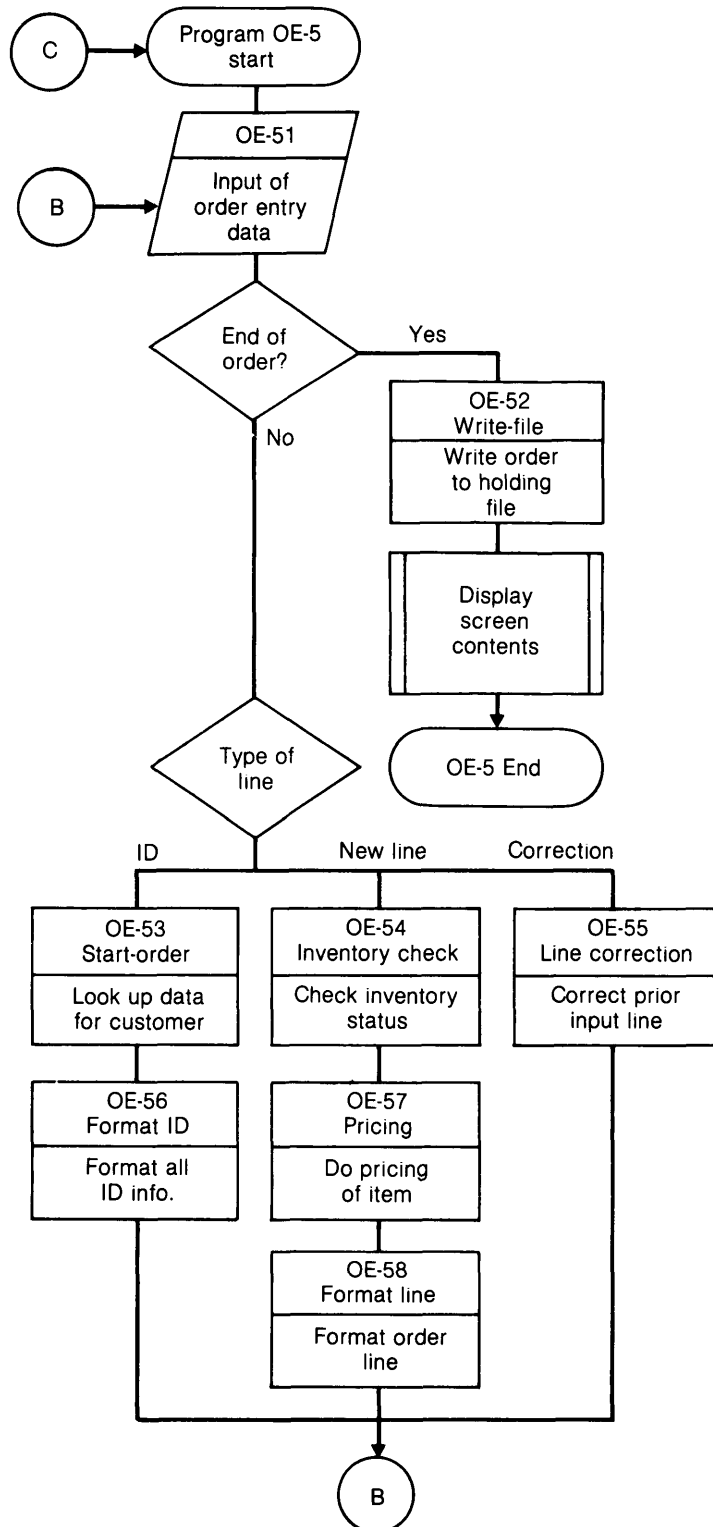


Figure B-3 illustrates flowcharting conventions. On the left is a system flowchart describing the flow of processing, files used, and so on. Major processing programs are described by striped processing symbols. Sets of program flowcharts are prepared for each of the processing routines. On the right is a summary program flowchart, each of the striped processing boxes refers to a routine described in more detail in the set of flowcharts.

(cont.)

Figure B-3 Flowcharting conventions





(cont.)

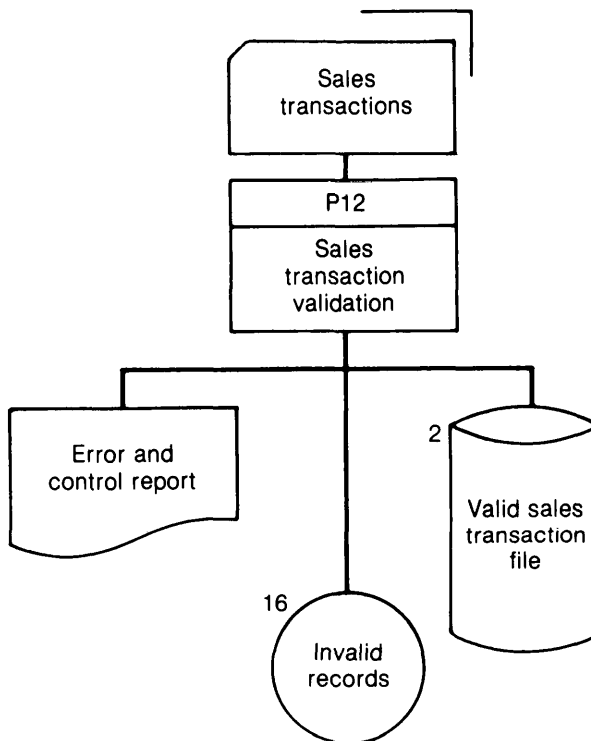
## Computer Application Flowcharts for Audit Analysis

The auditor may review client flowcharts or draw flowcharts for audit analysis as follows:

- 1 Review client system flowcharts to understand flow of the processing, input, output, files, and so on
- 2 Document client application controls by drawing a system control flowchart corresponding to the normal client system flowchart showing data flows (See fig B-4 )
- 3 Document application processing and controls both inside and outside EDP by use of a horizontal flowchart (See fig B-5 )

The system control flowchart is an extension of the data-flow-oriented system flowchart. Controls can be noted on the system flowchart, but it may be useful to draw a separate control flowchart to identify the flow of controls in an application. The control flowchart has the same outline as the data flow system flowchart, but the notation in or alongside each box defines the relevant controls. Such a control flowchart can be useful both in application design and in audit review. Figure B-4 shows a simple data flowchart and the corresponding control flowchart.

The typical computer application system flowchart flows from top to bottom and from left to right and concentrates on activities within data processing. The standard symbols and conventions can also be used in a horizontal flowchart to show the processing activities of all departments with respect to the application. An example is given in figure B-5.



**Figure B-4A** Data flowchart

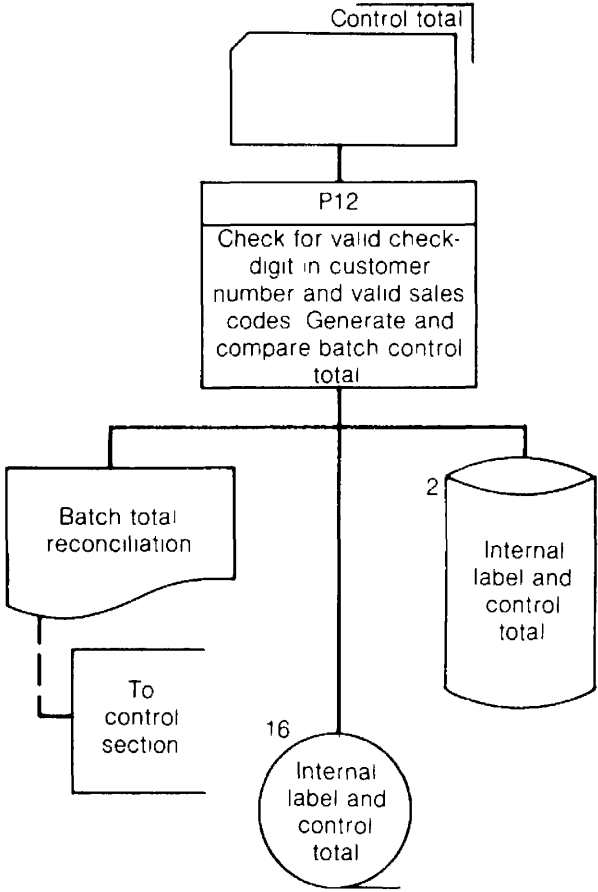
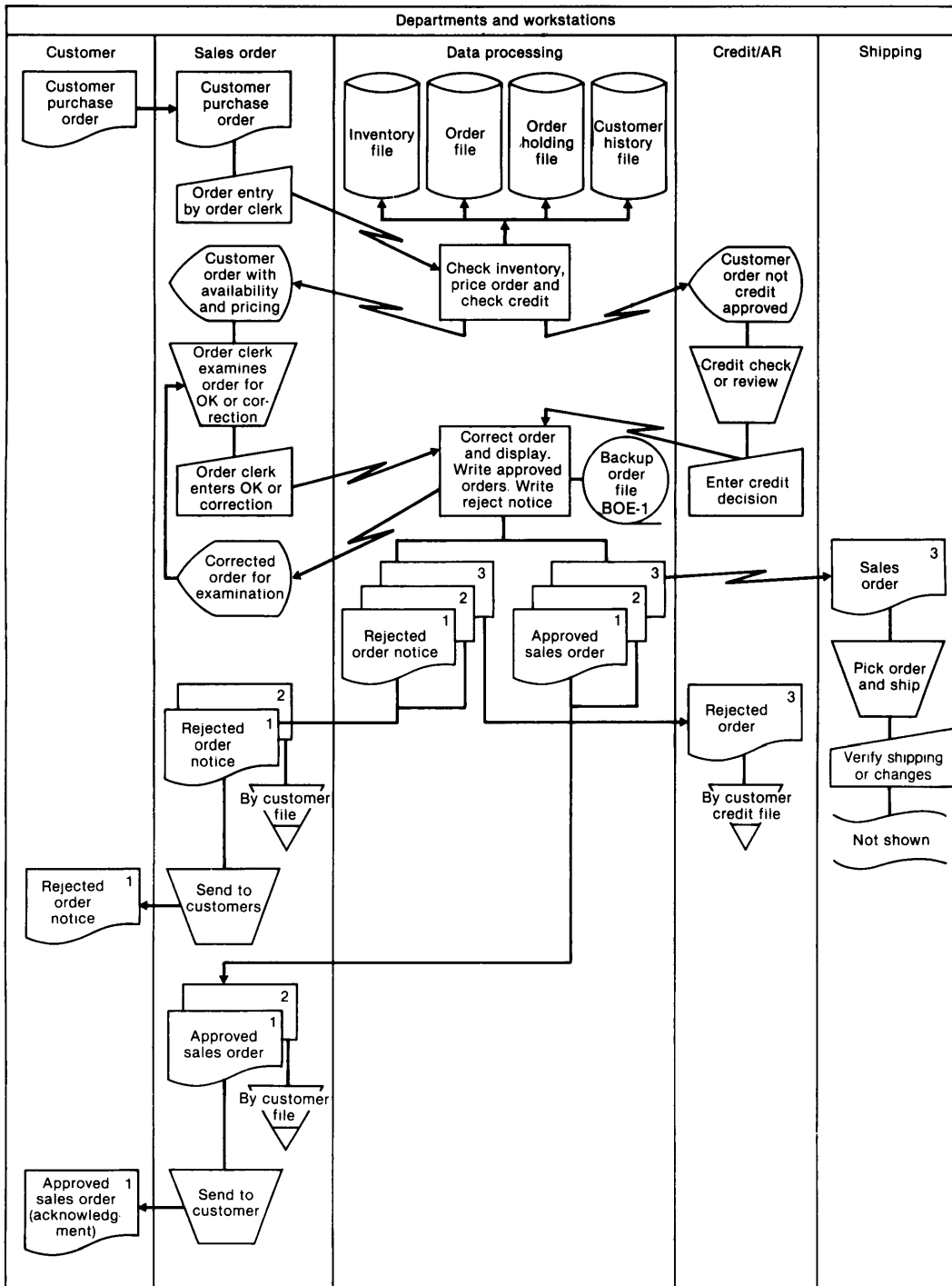


Figure B-4B Control flowchart

**Figure B-5** Sales order processing horizontal flowchart

Audit notes		
Controls		Control Weakness
1. All orders held on order holding file until approved. Daily list of orders still being held more than one day (see day-end reports application).		1. No control over customer orders until after entry. Loss or failure to enter not detected.
2. Control totals		2. No check digit on customer number but visual echo of customer name is used and appears adequate.
	Holding File	Order File
Orders & lines entered	x	
Orders & lines approved and put on order file	x	x
Dollars of approved orders		x
Number of orders and lines being held for credit approval	x	
Number of orders and lines on file	x	x
3. Management summary of reasons for rejections (see day-end report application).		3. No security code or record of credit department individual doing approval. Check only for credit department terminal input.
4. Pricing table maintained on inventory file. Control total on pricing table and all changes controlled. See chart on pricing.		4. No management summary or control analysis of credit decisions made by credit department.
5. Input validation on: Item code—echo display with item description  Item order quantity—reasonableness based on customer history		
6. Orders backed up for recovery purposes		







---

# An Example of Documentation



This appendix is an example of documentation for an application. It is illustrative of documentation practice, however, alternative approaches to documentation are also acceptable.

The example documentation is for an application that updates an inventory master file. The file is on disk storage with records accessed by random access procedures.

The relationship of the elements of comprehensive application documentation described in chapter 5 and this simple example are shown below.

<u>Documentation Element</u>	<u>Simple Example Documentation</u>
<i>System documentation</i>	<i>Not needed for simple system—program documentation sufficient</i>
<i>Job documentation</i>	<i>Not needed for simple system—program documentation sufficient</i>
<i>Program documentation</i>	
<i>Program narrative</i>	<i>Program narrative</i>
<i>Program flowcharts</i>	<i>Program flowcharts (sample of 8 from a set of 25 pages)</i>
<i>Source code listing</i>	<i>Program source code listing (sample of 6 pages extracted from full listing)</i>
<i>Test results</i>	<i>Not included</i>
<i>Change notice</i>	<i>Not included (no changes)</i>
<i>File documentation</i>	<i>File description (associated with program documentation rather than a separate data dictionary)</i>
<i>Operations documentation</i>	
<i>Job and program narratives</i>	<i>See program narrative.</i>
<i>Input/output chart</i>	<i>Not needed. Information included in program narrative</i>
<i>Job command language list</i>	<i>Not included because simple system</i>
<i>Error message list</i>	<i>Error message list</i>
<i>Restart and recovery instructions</i>	<i>Restart and recovery</i>
<i>User documentation</i>	<i>Not needed (no users outside of data processing)</i>

Some comments about each of the six elements of documentation used in this example will be useful in understanding the purpose and form of each part.

## **PROGRAM NARRATIVE—Figure C-1**

The three-page program narrative is also a system narrative, a job narrative, and a narrative for operations because the program narrative contains everything necessary for these other pur-

poses. The purpose of the program is described, and inputs, outputs, and flow of processing are summarized.

#### **PROGRAM FLOWCHARTS—Figure C-2**

As explained in the text, there are differences of opinion on the use of program flowcharts in documentation, but this appendix uses the traditional program flowcharts. The full set of program flowcharts is twenty-five pages; only eight sample pages corresponding to the pages of program listing are included. The flowcharts adhere to the American National Standard conventions (see Appendix B). The modules of the program (coded as program sections) are each diagrammed on a separate page.

#### **PROGRAM SOURCE CODE LISTING—Figure C-3**

The program contains over 600 lines of instructions in the COBOL programming language. For the purposes of illustration, only selected portions of the program are included. These show the first three divisions of a COBOL program (IDENTIFICATION, ENVIRONMENT, and DATA) and those parts of the PROCEDURE division flowcharted in Figure C-2. The program was prepared in an on-line programming environment, and the dates on each line of code were added by the system. The program design is modular, with each section being a separate program module. The program is not fully structured in that it uses GO TO instructions in the MAINLINE module. The program contains a number of examples of programmed controls. Note, for example, section 20 for computing control totals and section 40 for writing error messages on the Error & Control Report.

#### **FILE DESCRIPTION—Figure C-4**

There are descriptions for each of the two files, a transaction file and a master file. The items within the records in the file are described by names and COBOL pictures. A picture of 9(4) means a data item with storage for four numeric digits; ×(40) means a data item with storage for forty alphanumeric characters. The levels indicate the hierarchy of data. For example, the data item master-last-received-date has three subitems for year, month, and day.

#### **ERROR MESSAGE LIST—Figure C-5**

When an error occurs during the execution of the program, an error message is printed on an Error & Control report. The error message list explains, for each error message, the reason for (meaning) of the message and details the action to be taken.

#### **RESTART AND RECOVERY—Figure C-6**

This element of documentation explains what should be done in case the application is interrupted by a failure of hardware or software. The instructions in this case are very simple because it is more efficient to rerun the application than to have elaborate recovery procedures.

AICPA	Document Type PROGRAM NARRATIVE	Revision No Ø	Page 1	Of 3
System INVENTORY	Name UPDATE INVENTORY	Revision Date	Original Date 3/16/8X	
Subsystem	Number INPØ82	Analyst DLA	Design Level	
			General	Detail X

1. Purpose of the Program  

The program has three major functions:

Edit inventory master file update transactions to detect the following errors:

  - No master record exists for the transaction
  - Quantity to be issued exceeds the quantity on hand.
  - Transaction type code is not valid

Update the master file fields for quantity, date of last usage, and date of last receipt and write a new record on the master file.

Write an output report showing the details of all transactions that failed the edits and all control totals that are needed to balance inputs and outputs.
2. Inputs  

INFØ82 Inventory Transactions  
 INFØ83 Inventory Master
3. Outputs  

INRØ82 Error & Control Report  
 INFØ83 Inventory Master {updated}
4. Processing Requirements  

The functional processing takes place along the basic lines specified in paragraph 1, "Purpose of the Program."

  - {1} A record from the inventory transaction file {INFØ82} is read.
  - {2} Transaction part number and quantity are added to the overall control totals.

(cont.)

Figure C-1 Program narrative

AICPA	Document Type PROGRAM NARRATIVE	Revision No Ø	Page 2	Of 3
System INVENTORY	Name UPDATE INVENTORY	Revision Date	Original Date 3/16/8X	
Subsystem	Number INPØ82	Analyst DLA	Design Level	
			General	Detail X

- {3} If the transaction type code is not 1 {receipt} or 2 {withdrawal}, the transaction is invalid. An error message and the details of the transaction are printed on the Error & Control Report {INRØ82}.
- {4} The corresponding record from the inventory master file {INFØ83} is sought. If the master record cannot be found, an error message and the details of the transaction are printed on the Error & Control Report {INRØ82}. The part number and quantity are added to the reject control totals.
- {5} If the transaction is a withdrawal {type code 2} and the quantity to be withdrawn exceeds the quantity on the master file, the transaction will not be processed. An error message and the details of the transaction are printed on the Error & Control Report {INRØ82}. The part number and quantity are added to the reject control totals.
- {6} If the transaction is a receipt {type code 1}:
  - {a} Transaction quantity is added to master file quantity.
  - {b} Transaction date is moved to date of last receipt on the master file.
- {7} If the transaction is a withdrawal {type code 2}:
  - {a} Transaction quantity is subtracted from master file quantity.
  - {b} Transaction date is moved to date of last usage on the master file.
- {8} The part number and quantity are added to update control totals.
- {9} An updated master file {INFØ83} record is written.
- {10} At the end of the inventory transaction file {INFØ82}, reject control totals and update control totals are printed on the Error & Control Report {INRØ82}.

AICPA		Document Type	Revision No	Page	Of
		PROGRAM NARRATIVE	Ø	3	3
System	Name		Revision Date	Original Date	
INVENTORY	UPDATE INVENTORY			3/16/8X	
Subsystem		Number	Analyst	Design Level	
		INPØ82	DLA	General	Detail
					X

{11} The two sets of control totals are added together to produce combined totals, which are also printed. These combined totals are compared with the overall control totals {see step 2}. The two sets of control totals should agree. If they do not, a logical or processing error took place during the job. A message, indicating whether or not the two sets of controls are in balance, will be printed out at the end of the Error & Control Report {INRØ82}.

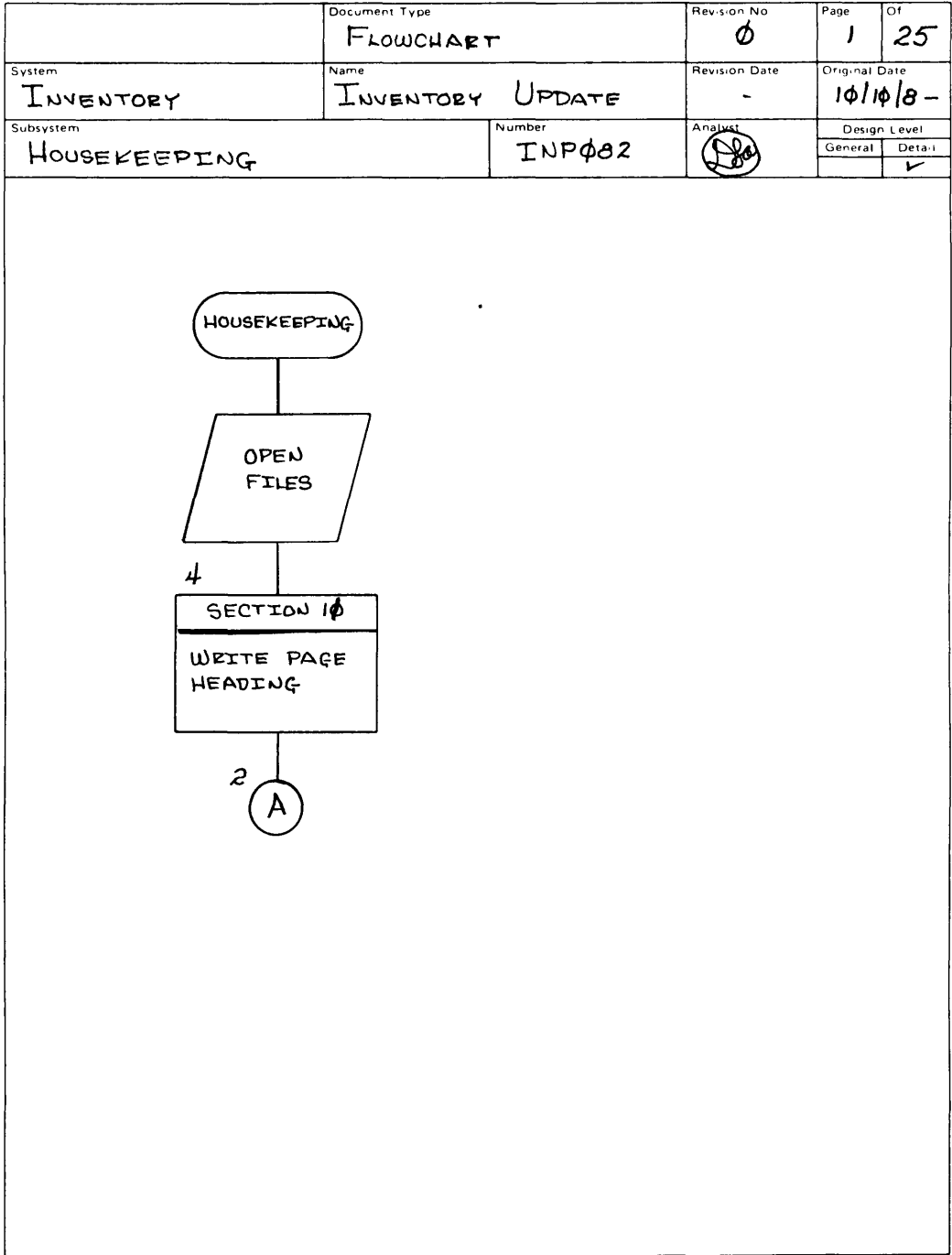

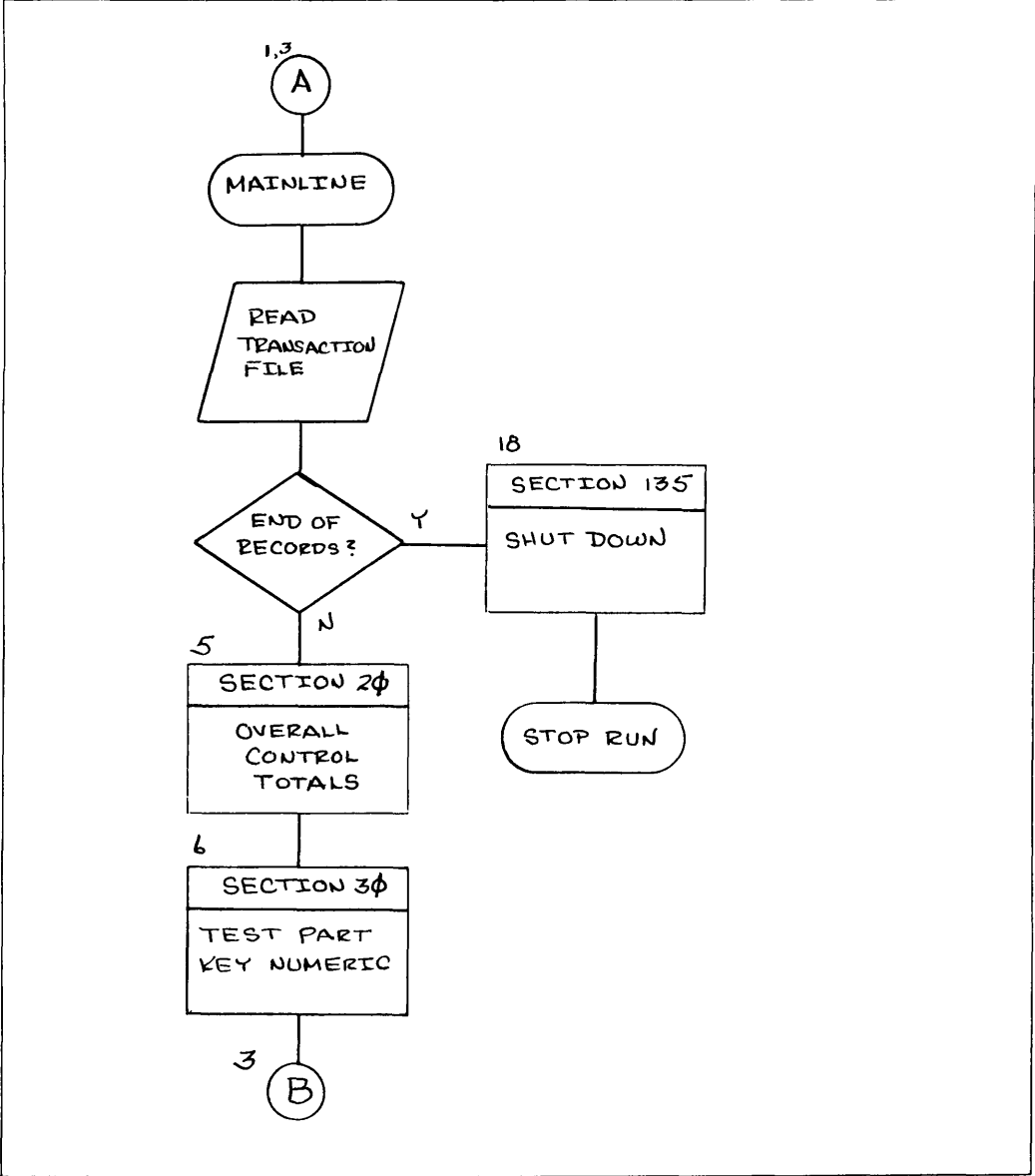


Figure C-2 Program flowchart—Program HOUSEKEEPING SECTION

	Document Type <b>FLOWCHART</b>	Revision No <b>0</b>	Page <b>2</b>	Of <b>25</b>
System <b>INVENTORY</b>	Name <b>INVENTORY UPDATE</b>	Revision Date <b>-</b>	Original Date <b>10/10/8-</b>	
Subsystem <b>MAINLINE</b>		Number <b>INP082</b>	Analyst 	
			Design Level	
			General	Detail
				<input checked="" type="checkbox"/>



(cont.)

Figure C-2 Program flowchart—Program MAINLINE SECTION

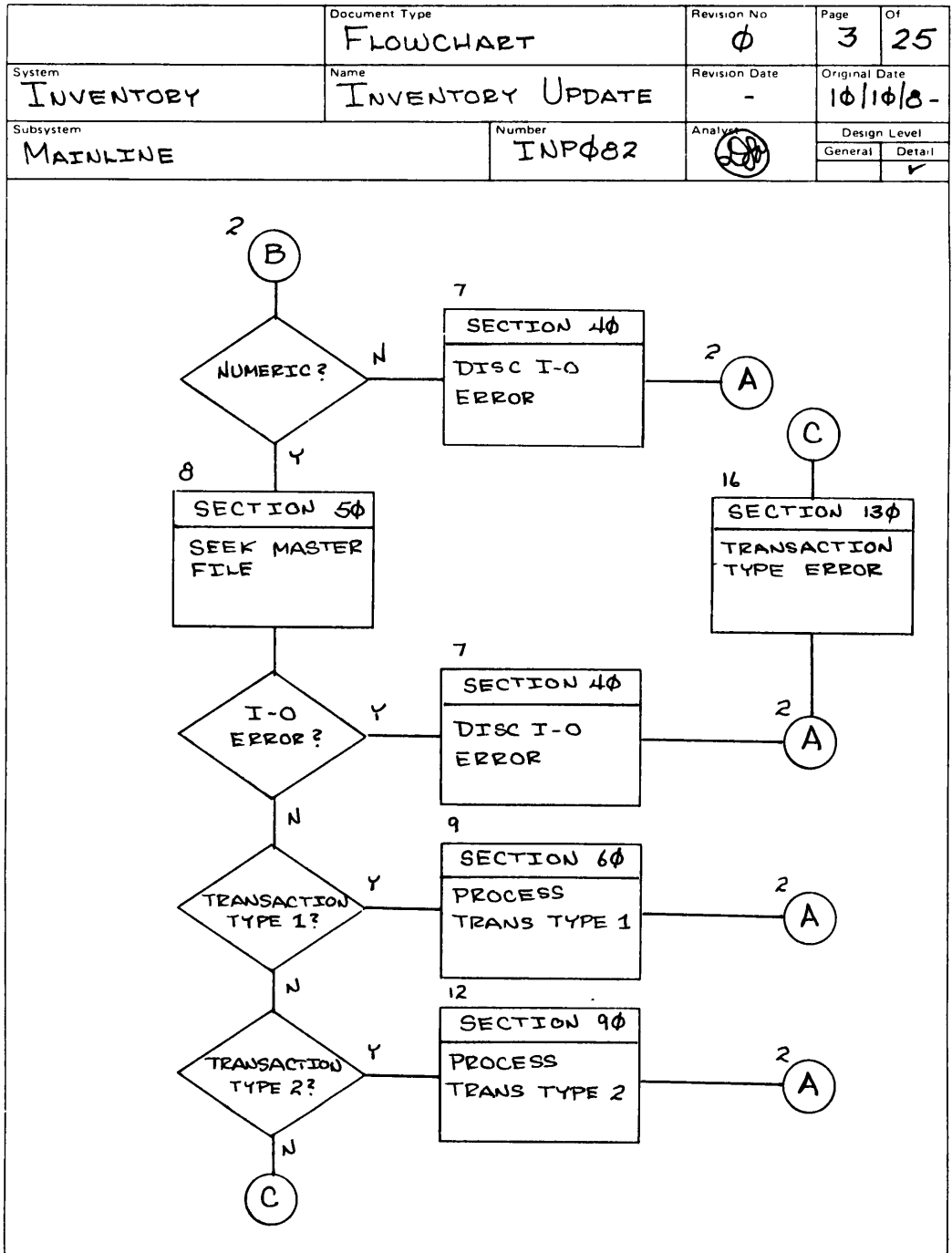


Figure C-2 (cont.)



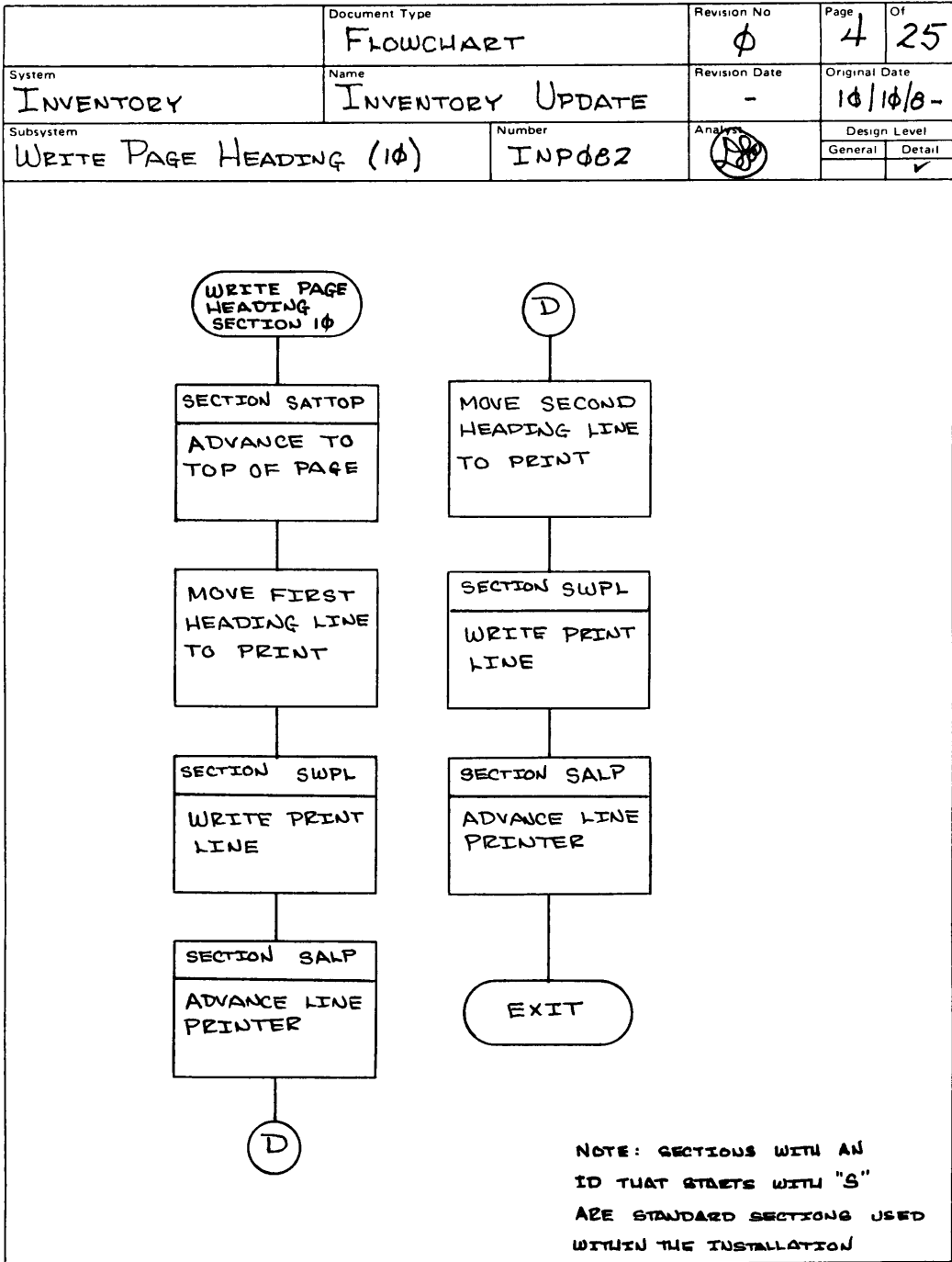
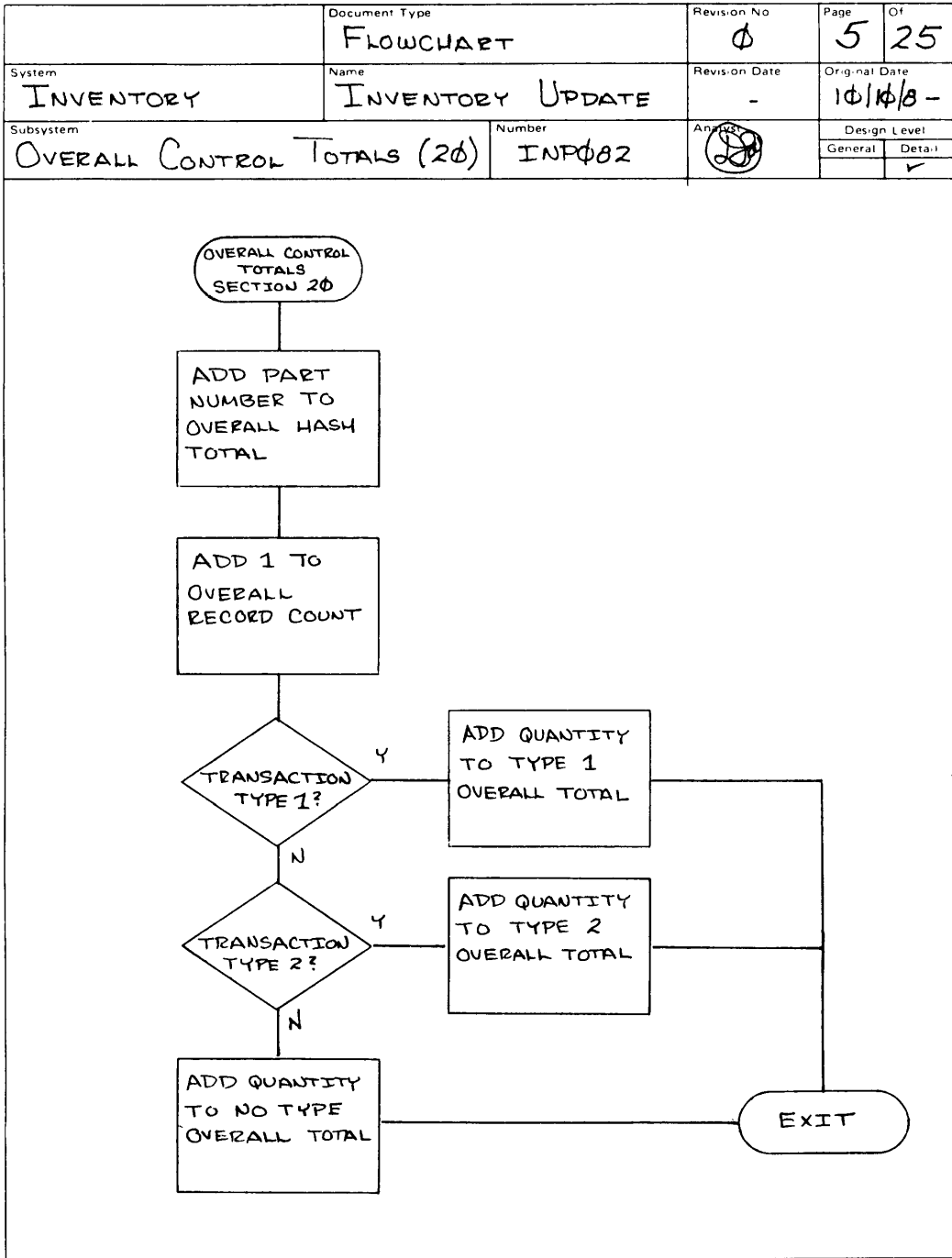
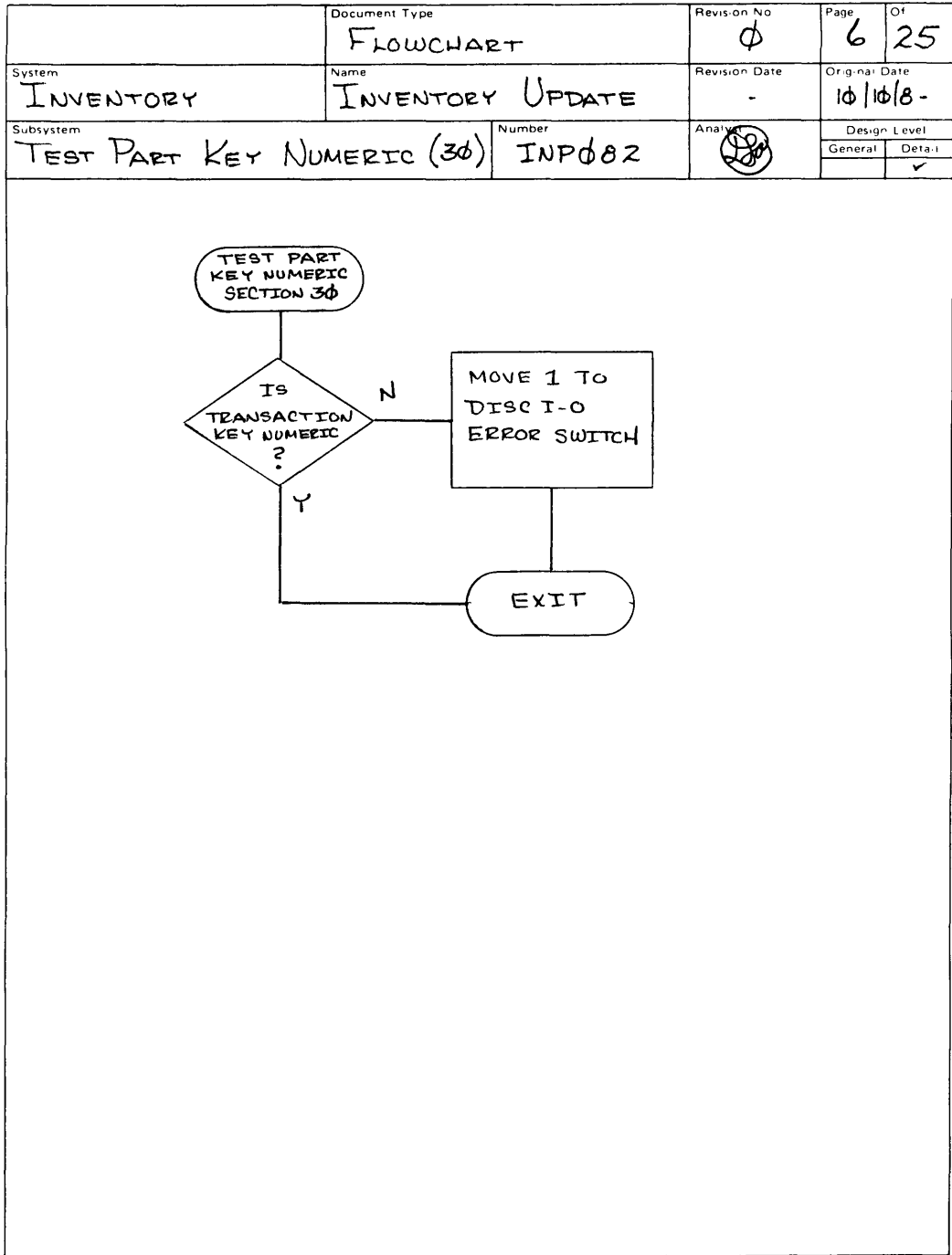


Figure C-2 Program flowchart—Program WRITE-PAGE-HEADING SECTION



**Figure C-2** Program flowchart—Program OVERALL-CONTROL-TOTALS SECTION



**Figure C-2** Program flowchart—Program TEST-PART-KEY-NUMERIC SECTION

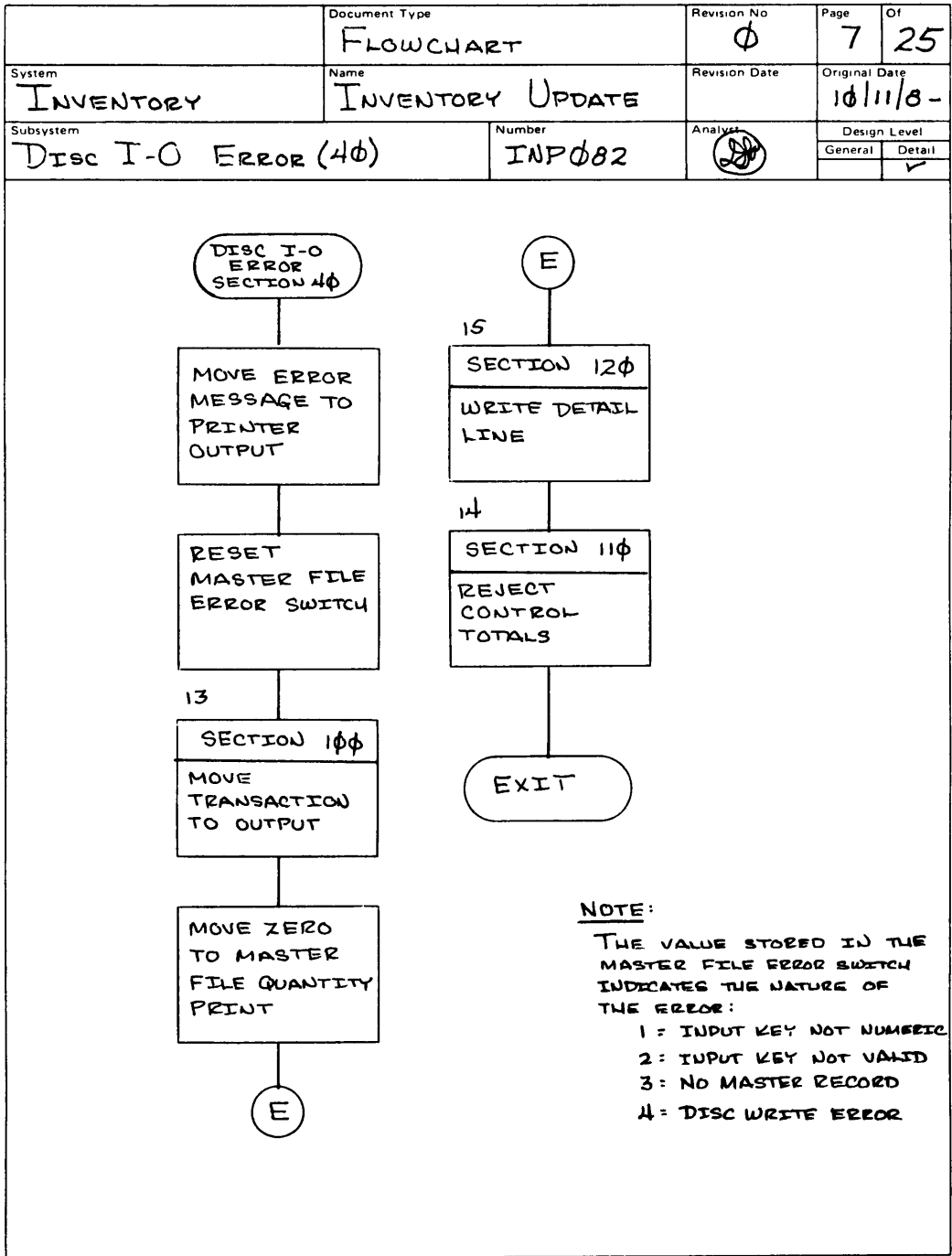


Figure C-2 Program flowchart—Program DISC-I-O-ERROR SECTION

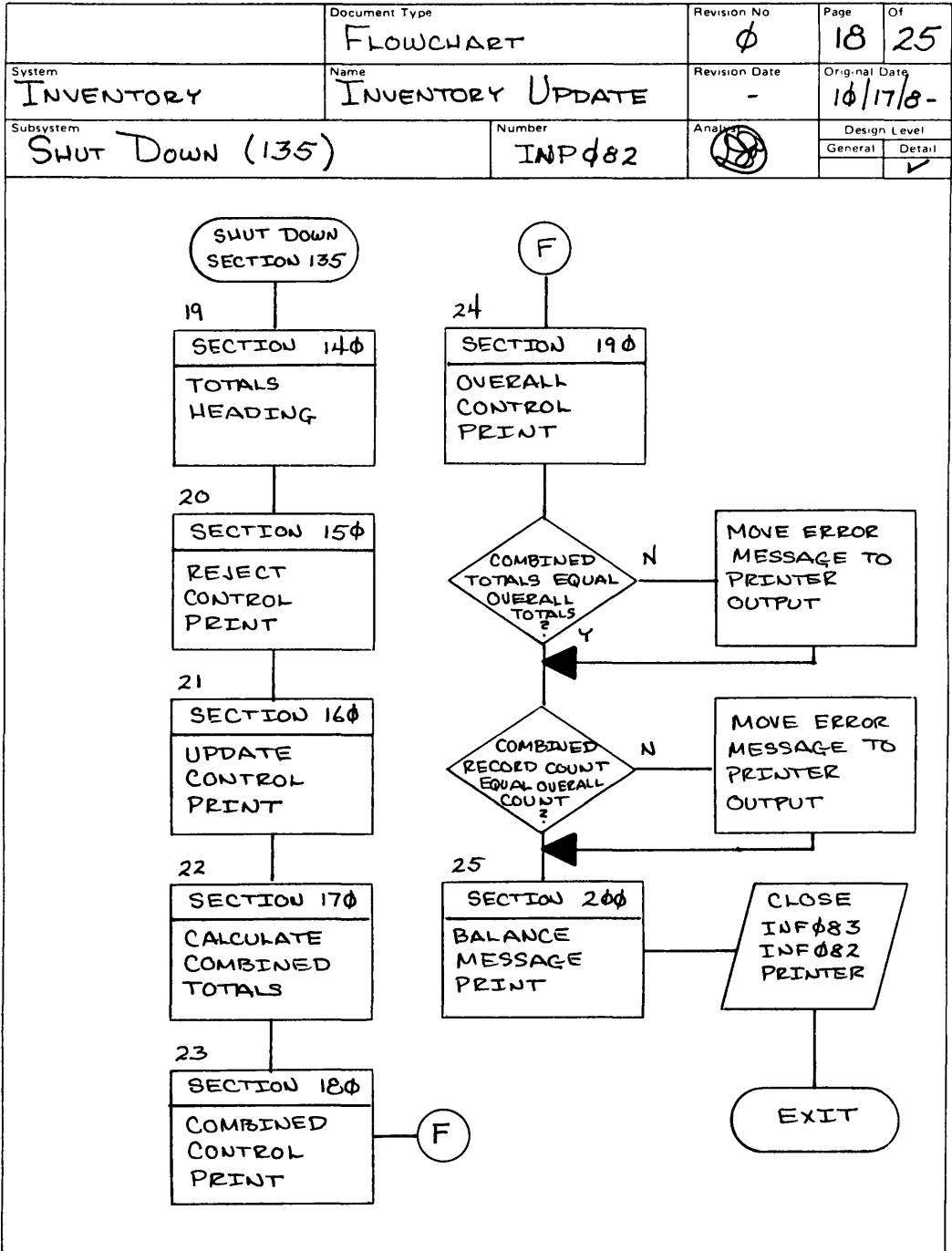


Figure C-2 Program flowchart—Program SHUT-DOWN SECTION

```

000100 IDENTIFICATION DIVISION. 03/21/83
000200 PROGRAM-ID. INP082. 03/21/83
000300 AUTHOR. DONALD L ADAMS. 03/21/83
000400 INSTALLATION. 03/21/83
000500 AICPA. 03/21/83
000600 DATE-WRITTEN. 03/21/83
000700 1 JAN 198X. 03/21/83
000800 DATE-COMPILED. 03/21/83
                                TIME 10:56 DATE 04/21/83.
000900 SECURITY. 03/21/83
001000 BE ADVISED THAT ALL COMPUTER PROGRAMS AND ASSOCIATED 03/21/83
001100 MATERIAL PREPARED FOR THE AICPA ARE ITS PROPERTY - 03/21/83
001200 WITHOUT WRITTEN CONSENT, GIVEN BY CONTRACT OR SOME 03/21/83
001300 OTHER MANNER, NO DOCUMENT OR ASSOCIATED MATERIAL 03/21/83
001400 CONCERNED WITH THE DESIGN, DEVELOPMENT, OR IMPE- 03/21/83
001500 MENTATION OF AICPA COMPUTER PROGRAMS MAY BE COPIED, 03/21/83
001600 REPRINTED, OR REPRODUCED IN ANY MATERIAL FORM, 03/21/83
001700 EITHER WHOLLY OR IN PART, NOR MAY ITS CONTENTS, 03/21/83
001800 OR ANY METHOD OR TECHNIQUE AVAILABLE THEREFROM, BE 03/21/83
001900 DISCLOSED, EITHER WHOLLY OR IN PART, TO ANY OTHER 03/21/83
002000 PERSON WHATSOEVER. 03/21/83
002100 ENVIRONMENT DIVISION. 03/21/83
002200 CONFIGURATION SECTION. 03/21/83
002300 SOURCE-COMPUTER. B-2900. 03/21/83
002400 OBJECT-COMPUTER. B-2900. 03/21/83
002500 INPUT-OUTPUT SECTION. 03/21/83
002600 FILE-CONTROL. 03/21/83
002700 SELECT INF083 ASSIGN TO DISKPACK 03/21/83
002800 RESERVE 2 ALTERNATE AREAS 03/21/83
002900 ACCESS MODE IS RANDOM 03/21/83
003000 ACTUAL KEY IS PART-KEY. 03/21/83
003100 SELECT INF082 ASSIGN TO TAPE. 03/21/83
003200 SELECT S-PRINT-FILE ASSIGN TO PRINTER. 03/21/83

```

Figure C-3 Program source code listing—IDENTIFICATION and ENVIRONMENT DIVISIONS

```

003250 / 03/21/83
003300 DATA DIVISION. 03/21/83
003400 FILE SECTION. 03/21/83
003500 FD INF083 03/21/83
003600 FILE CONTAINS 1 BY 10000 RECORDS. 03/21/83
003700 01 INVENTORY-MASTER-RECORD. 03/21/83
003800 05 MASTER-PART-NUMBER PIC 9(4). 03/21/83
003900 05 MASTER-PART-DESCRIPTION PIC X(40). 03/21/83
004000 05 MASTER-UNIT-PRICE PIC 9(4)V9(2). 03/21/83
004100 05 MASTER-QTY-ON-HAND PIC 9(6). 03/21/83
004200 05 MASTER-LAST-USAGE-DATE. 03/21/83
004300 10 MASTER-LAST-USE-YR PIC 9(2). 03/21/83
004400 10 MASTER-LAST-USE-MONTH PIC 9(2). 03/21/83
004500 10 MASTER-LAST-USE-DAY PIC 9(2). 03/21/83
004600 05 MASTER-LAST-RECEIVED-DATE. 03/21/83
004700 10 MASTER-LAST-RECD-YR PIC 9(2). 03/21/83
004800 10 MASTER-LAST-RECD-MONTH PIC 9(2). 03/21/83
004900 10 MASTER-LAST-RECD-DAY PIC 9(2). 03/21/83
005000 FD INF082 03/21/83
005100 BLOCK CONTAINS 10 RECORDS 03/21/83
005200 DATA RECORD IS INVENTORY-TRANSACTION-RECORD. 03/21/83
005300 01 INVENTORY-TRANSACTION-RECORD. 03/21/83
005400 05 TRANS-PART-NUMBER PIC 9(4). 03/21/83
005500 05 TRANS-TYPE-CODE PIC 9(1). 03/21/83
005600 05 TRANS-QTY PIC 9(6). 03/21/83
005700 05 TRANS-DATE PIC 9(6). 03/21/83
005750 05 FILLER PIC X(3). 03/21/83
005800 FD S-PRINT-FILE 03/21/83
005900 DATA RECORD IS S-PRINT-RECORD. 03/21/83
006000 01 S-PRINT-RECORD PIC X(132). 03/21/83

```

Figure C-3 Program source code listing—DATA DIVISION

019100 /		03/21/83
019200	PROCEDURE DIVISION.	03/21/83
019300 *		03/21/83
019400	HOUSEKEEPING SECTION.	03/21/83
019500 *		03/21/83
019600	MSK0100.	03/21/83
019700	OPEN I-O INFO83.	03/21/83
019800	OPEN INPUT INFO82.	03/21/83
019900	OPEN OUTPUT S-PRINT-FILE.	03/21/83
020000	PERFORM WRITE-PAGE-HEADING.	03/21/83
020100 *		03/21/83
020200	MAINLINE SECTION.	03/21/83
020300 *		03/21/83
020400	MAIN0100.	03/21/83
020500	READ INFO82	03/21/83
020600	AT END PERFORM SHUT-DOWN	03/21/83
020700	STOP RUN.	03/21/83
020800	PERFORM OVERALL-CONTROL-TOTALS.	03/21/83
020900	PERFORM TEST-PART-KEY-NUMERIC.	03/21/83
021000	IF DISC-I-O-ERROR-SWITCH EQUALS "1"	03/21/83
021100	PERFORM DISC-I-O-ERROR	03/21/83
021200	GO TO MAIN0100.	03/21/83
021300	PERFORM SEEK-MASTER-FILE.	03/21/83
021400	IF DISC-I-O-ERROR-SWITCH NOT EQUAL "0"	03/21/83
021500	PERFORM DISC-I-O-ERROR	03/21/83
021600	GO TO MAIN0100.	03/21/83
021700	IF TRANS-TYPE-CODE EQUALS 1	03/21/83
021800	PERFORM PROCESS-TRANS-TYPE-1	03/21/83
021900	GO TO MAIN0100.	03/21/83
022000	IF TRANS-TYPE-CODE EQUALS 2	03/21/83
022100	PERFORM PROCESS-TRANS-TYPE-2	03/21/83
022200	GO TO MAIN0100.	03/21/83
022300	PERFORM TRANSACTION-TYPE-ERROR.	03/21/83
022400	GO TO MAIN0100.	03/21/83

**Figure C-3** Program source code listing—PROCEDURE DIVISION (HOUSEKEEPING and MAINLINE SECTIONS)

022500 /		03/21/83
022600	WRITE-PAGE-HEADING SECTION.	03/21/83
022700 *		03/21/83
022800 *	SECTION 10	03/21/83
022900 *		03/21/83
023000	S0010.	03/21/83
023100	PERFORM S-ADVANCE-TO-TOP-OF-PAGE.	03/21/83
023200	MOVE HL04 TO S-PRINT-RECORD.	03/21/83
023300	PERFORM S-WRITE-PRINT-LINE.	03/21/83
023400	PERFORM S-ADVANCE-LINE-PRINTER.	03/21/83
023500	MOVE HL01 TO S-PRINT-RECORD.	03/21/83
023600	PERFORM S-WRITE-PRINT-LINE.	03/21/83
023700	PERFORM S-ADVANCE-LINE-PRINTER.	03/21/83
023800	S0010-EXIT.	03/21/83
023900	EXIT.	03/21/83
024000 *		03/21/83
024100	OVERALL-CONTROL-TOTALS SECTION.	03/21/83
024200 *		03/21/83
024300 *	SECTION 20	03/21/83
024400 *		03/21/83
024500	S0020.	03/21/83
024600	ADD TRANS-PART-NUMBER TO A16-OVERALL-HASH.	03/21/83
024700	ADD 1 TO A20-OVERALL-RECORD-COUNT.	03/21/83
024800	IF TRANS-TYPE-CODE EQUAL 1	03/21/83
024900	ADD TRANS-QTY TO A17-OVERALL-TYPE-1	03/21/83
025000	ELSE	03/21/83
025100	IF TRANS-TYPE-CODE EQUAL 2	03/21/83
025200	ADD TRANS-QTY TO A18-OVERALL-TYPE-2	03/21/83
025300	ELSE	03/21/83
025400	ADD TRANS-QTY TO A19-OVERALL-NO-TYPE.	03/21/83
025500	S0020-EXIT.	03/21/83
025600	EXIT.	03/21/83
025700 *		03/21/83
025800	TEST-PART-KEY-NUMERIC SECTION.	03/21/83
025900 *		03/21/83
026000 *	SECTION 30	03/21/83
026100 *		03/21/83
026200	S0030.	03/21/83
026300	IF TRANS-PART-NUMBER NOT NUMERIC	03/21/83
026400	MOVE "1" TO DISC-I-O-ERROR-SWITCH.	03/21/83
026500	S0030-EXIT.	03/21/83
026600	EXIT.	03/21/83

**Figure C-3** Program source code listing—WRITE-PAGE-HEADING, OVERALL-CONTROL-TOTALS, and TEST-PART-KEY-NUMERIC SECTIONS



026700 /		03/21/83
026800	DISC-I-O-ERROR SECTION.	03/21/83
026900 *		03/21/83
027000 *	SECTION 40	03/21/83
027100 *		03/21/83
027200	S0040.	03/21/83
027300	IF DISC-I-O-ERROR-SWITCH EQUAL "1"	03/21/83
027400	MOVE EM-070 TO DL01-010-MESSAGE	03/21/83
027500	ELSE	03/21/83
027600	IF DISC-I-O-ERROR-SWITCH EQUAL "2"	03/21/83
027700	MOVE EM-010 TO DL01-010-MESSAGE	03/21/83
027800	ELSE	03/21/83
027900	IF DISC-I-O-ERROR-SWITCH EQUAL "3"	03/21/83
028000	MOVE EM-080 TO DL01-010-MESSAGE	03/21/83
028100	ELSE	03/21/83
028200	IF DISC-I-O-ERROR-SWITCH EQUAL "4"	03/21/83
028300	MOVE EM-060 TO DL01-010-MESSAGE	03/21/83
028400	ELSE	03/21/83
028500	MOVE EM-090 TO DL01-010-MESSAGE.	03/21/83
028600	S0041.	03/21/83
028700	MOVE "0" TO DISC-I-O-ERROR-SWITCH.	03/21/83
028800	PERFORM MOVE-TRANSACTION-TO-OUTPUT.	03/21/83
028900	MOVE ZERO TO DL01-060-MAST-QTY.	03/21/83
029000	PERFORM WRITE-DETAIL-LINE.	03/21/83
029100	PERFORM REJECT-CONTROL-TOTALS.	03/21/83
029200	S0040-EXIT.	03/21/83
029300	EXIT.	03/21/83

Figure C-3 Program source code listing—DISC-I-O-ERROR SECTION

042800 /		03/21/83
042900	SHUT-DOWN SECTION.	03/21/83
043000 *		03/21/83
043100 *	SECTION 135	03/21/83
043200 *		03/21/83
043300	S0135.	03/21/83
043400	PERFORM TOTALS-HEADING.	03/21/83
043500	PERFORM REJECT-CONTROL-PRINT.	03/21/83
043600	PERFORM UPDATE-CONTROL-PRINT.	03/21/83
043700	PERFORM CALCULATE-COMBINED-TOTALS.	03/21/83
043800	PERFORM COMBINED-CONTROL-PRINT.	03/21/83
043900	PERFORM OVERALL-CONTROL-PRINT.	03/21/83
044000	MOVE EM-040 TO FLO2-010-MESSAGE.	03/21/83
044100	IF A11-COMBINED-HASH NOT EQUAL A16-OVERALL-HASH	03/21/83
044200	MOVE EM-050 TO FLO2-010-MESSAGE.	03/21/83
044300	IF A12-COMBINED-TYPE-1 NOT EQUAL A17-OVERALL-TYPE-1	03/21/83
044400	MOVE EM-050 TO FLO2-010-MESSAGE.	03/21/83
044500	IF A13-COMBINED-TYPE-2 NOT EQUAL A18-OVERALL-TYPE-2	03/21/83
044600	MOVE EM-050 TO FLO2-010-MESSAGE.	03/21/83
044700	IF A14-COMBINED-NO-TYPE NOT EQUAL A19-OVERALL-NO-TYPE	03/21/83
044800	MOVE EM-050 TO FLO2-010-MESSAGE.	03/21/83
044900	IF A15-COMBINED-RECORD-COUNT NOT EQUAL	03/21/83
045000	A20-OVERALL-RECORD-COUNT	03/21/83
045100	MOVE EM-050 TO FLO2-010-MESSAGE.	03/21/83
045200	PERFORM BALANCE-MESSAGE-PRINT.	03/21/83
045300	CLOSE INFO83 WITH RELEASE.	03/21/83
045400	CLOSE INFO82.	03/21/83
045500	CLOSE S-PRINT-FILE.	03/21/83
045600	S0135-EXIT.	03/21/83
045700	EXIT.	03/21/83

Figure C-3 Program source code listing—SHUT-DOWN SECTION

AICPA		Document Type FILE DESCRIPTION	Revision No Ø	Page 1	Of 1
System INVENTORY		Name INVENTORY TRANSACTION FILE	Revision Date	Original Date 3/17/8X	
Subsystem		Number INFØ82	Analyst DLA	Design Level	
				General	Detail
					X
Level	Name	Picture	Comments		
Ø1	Inventory-transaction-record				
Ø5	Trans-part-number	9{4}			
Ø5	Trans-type-code	9{1}	1 = receipt 2 = withdraw		
Ø5	Trans-gty	9{6}			
Ø5	Trans-date	9{6}			

**Figure C-4** File description—Inventory transaction file

AICPA		Document Type FILE DESCRIPTION	Revision No Ø	Page 1	Of 1
System INVENTORY		Name INVENTORY MASTER FILE	Revision Date	Original Date 3/17/8X	
Subsystem			Number INFØ83	Analyst DLA	Design Level General Detail X
Level	Name	Picture	Comments		
Ø1	Inventory-master record				
Ø5	Master-part-number	9{4}			
Ø5	Master-part-description	X{4Ø}			
Ø5	Master-unit-price	9{4} V 9{2}			
Ø5	Master-qty-on-hand	9{6}			
Ø5	Master-last-usage date				
1Ø	Master-last-use-yr	9{2}			
1Ø	Master-last-use-month	9{2}			
1Ø	Master-last-use-day	9{2}			
Ø5	Master-last-received-date				
1Ø	Master-last-recvd-yr	9{2}			
1Ø	Master-last-recvd-month	9{2}			
1Ø	Master-last-recvd-day	9{2}			

Figure C-4 File description—Inventory master file

AICPA		Document Type	Revision No	Page	Of
		ERROR MESSAGE LIST	Ø	1	2
System		Name	Revision Date	Original Date	
INVENTORY		UPDATE INVENTORY		3/16/8X	
Subsystem		Number	Analyst	Design Level	
		INPØ82	DLA	General	Detail
					X
Message	Meaning	Action required			
No master record on file	There is no master record for the part number in the transaction	Control group research and correct. Either transaction part number was in error or a new master record must be added to the file.			
@ty issued exceeds on hand qty.	The transaction calls for the withdrawal of an inventory quantity in excess of the amount shown as available on the master file	Control group research and correct. Either quantity on the master file must be adjusted upward or the quantity on the transaction must be reduced. It is also possible the transaction type code should have been 1 {receipt} rather than 2 {withdrawal}			
Invalid transaction type code	The transaction type code is not 1 or 2	Control group research and correct. Type code must be either 1 or 2.			
Combined totals and overall totals do not balance	Control totals on transaction part numbers and quantities and/or the number of transactions read from the input file {INPØ82} do not agree with control totals for transactions processed.	This error message should not occur. It indicates a processing error or a program fault in program logic. The control group should contact systems for prompt research into, and correction of, the error.			

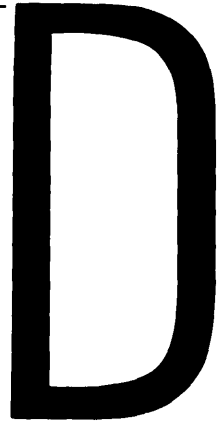
Figure C-5 Error message list

AICPA		Document Type	Revision No	Page	Of
INVENTORY <th>ERROR MESSAGE LIST</th> <td>Ø</td> <td>2</td> <td>2</td>		ERROR MESSAGE LIST	Ø	2	2
System	Name	Revision Date	Original Date		
UPDATE INVENTORY <td></td> <td colspan="3">3/16/8X</td>			3/16/8X		
Subsystem	Number	Analyst	Design Level		
	INPØ82	DLA	General	Detail	X
Key error on disc output	An attempt to write a record on disc failed because of an invalid value assigned to the record key.	Same as above			
Input key not numeric	The input file contains an error in the part number field of a transaction	Control group research and submit a correction			
Error type code is invalid	The program encountered an invalid error code.	This error message should not occur. It indicates a programming error. The control group should refer the problem to the systems group for correction.			

AICPA		Document Type	Revision No	Page	Of
		RESTART AND RECOVERY	Ø	1	1
System	INVENTORY	Name	Revision Date	Original Date	
		UPDATE INVENTORY		3/16/8X	
Subsystem		Number	Analyst	Design Level	
		INPØ82	DLA	General	Detail
					X
<p>1. Purpose                  -----</p> <p>In the worst case, this update program will run less than one hour from start to finish. As a result, no elaborate restart and recovery provisions are required. If a restart is required, the job will be rerun.</p> <p>2. Method                  -----</p> <p>{1} Using the system disc-to-tape utility, dump the partially updated inventory master file {INFØ83} from disc to tape. It is unlikely that this file will be needed, but it should be held until the restart and recovery has been completed.</p> <p>{2} Scratch the partially updated inventory master file {INFØ83} on the disc.</p> <p>{3} Load the beginning inventory master file, as it existed prior to the update processing, to disc from the most recent copy of the daily master back-up tape {INPØ84}.</p> <p>{4} Rerun the update inventory master program {INPØ82}.</p> <p>{5} If another failure occurs, the process should be repeated one more time. After three consecutive failures, contact the Operations Manager and report the problem. Do not attempt any additional processing.</p>					

Figure C-6 Restart and recovery

# Checklist for Evaluating the Audit Trail in a Computer Data Processing Application



Date \_\_\_\_ / \_\_\_\_ / \_\_\_\_

Application \_\_\_\_\_

## Functions performed

- Processing original source documents or direct terminal input

Describe \_\_\_\_\_

- Producing transaction documents

Describe \_\_\_\_\_

- Updating database, master file records, or stored tables used in processing

Describe \_\_\_\_\_

- Producing data for journal voucher, payment voucher, etc

Describe \_\_\_\_\_

## Input

## Source

1 \_\_\_\_\_

2 \_\_\_\_\_

3 \_\_\_\_\_

**390 Appendix D**

Files	Types
1 _____	_____
2 _____	_____
3 _____	_____
Output	Copies to
1 _____	_____
2 _____	_____
3 _____	_____

General Description of Processing/Audit Trail in Application

- 1 Describe how each output can be traced back to the source document or the set of source documents used in producing the output
  
- 2 Describe how processing of source documents can be traced to results or output
  
- 3 Describe the audit trail, where applicable, for—
  - a Triggered transactions
  
  - b Changes in database, master file records, or stored tables
  
  - c Analyses or results of complex computations

Transaction Source Record

- 1 Are there provisions for recording and identifying the source of each transaction?

	<u>Yes</u>	<u>No</u>
a Source document stored in batch	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
b Source document stored by some characteristic	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
c Multiple copies of source document stored in several files	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
_____		
_____		



	<u>Yes</u>	<u>No</u>
<i>d</i> Terminal input source listing	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
<i>e</i> Source records stored on a magnetic medium	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
<b>2</b> Are there adequate audit trail data items in the source record for the following items?		
<i>a</i> Identification of preparer	<input type="checkbox"/>	<input type="checkbox"/>
Describe. _____		
<i>b</i> Identification of approval	<input type="checkbox"/>	<input type="checkbox"/>
Describe. _____		
<i>c</i> Type of transaction	<input type="checkbox"/>	<input type="checkbox"/>
<i>d</i> Record or account affected	<input type="checkbox"/>	<input type="checkbox"/>
<i>e</i> Amount	<input type="checkbox"/>	<input type="checkbox"/>
<i>f</i> Processing reference	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
<i>g</i> Text explaining transaction (if relevant)	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		

Reference and Control List

	<u>Yes</u>	<u>No</u>
<b>1</b> Are there adequate provisions for reference and control list?	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
<i>a</i> List by physical batch of transactions	<input type="checkbox"/>	<input type="checkbox"/>
<i>b</i> List by logical batch characteristics	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
_____		
<b>2</b> Is the reference and control list used as a special journal?	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		

392 Appendix D

	<u>Yes</u>	<u>No</u>
3 Are there adequate audit trail data in the reference and control list?	<input type="checkbox"/>	<input type="checkbox"/>
<i>a</i> Source reference	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
<i>b</i> Account or record	<input type="checkbox"/>	<input type="checkbox"/>
<i>c</i> Amount	<input type="checkbox"/>	<input type="checkbox"/>
<i>d</i> Explanatory text (if relevant)	<input type="checkbox"/>	<input type="checkbox"/>
<i>e</i> Subtotals and totals	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		

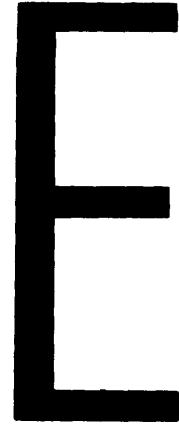
Process Tracing Data	<u>Yes</u>	<u>No</u>
1 Identified needs for tracing data		
<i>a</i> _____		
<i>b</i> _____		
<i>c</i> _____		
2 Are there adequate data to allow an investigator to trace, replicate, or reverse the processing?	<input type="checkbox"/>	<input type="checkbox"/>
<i>a</i> Processing understood without documentation	<input type="checkbox"/>	<input type="checkbox"/>
<i>b</i> Processing documentation	<input type="checkbox"/>	<input type="checkbox"/>
<i>c</i> Balances and transactions to support triggered transaction	<input type="checkbox"/>	<input type="checkbox"/>
<i>d</i> Items used in producing a balance or sum	<input type="checkbox"/>	<input type="checkbox"/>
<i>e</i> Data to recompute or estimate a computed result	<input type="checkbox"/>	<input type="checkbox"/>
<i>f</i> Data to identify change to database, master file, record, or stored tables		
Preparer	<input type="checkbox"/>	<input type="checkbox"/>
Time	<input type="checkbox"/>	<input type="checkbox"/>
Program version	<input type="checkbox"/>	<input type="checkbox"/>
Change	<input type="checkbox"/>	<input type="checkbox"/>
Before and after images	<input type="checkbox"/>	<input type="checkbox"/>

Transaction Source References	<u>Yes</u>	<u>No</u>
1 Are adequate source references available in transaction documents reports and accounts?		
a Physical batch	<input type="checkbox"/>	<input type="checkbox"/>
b Reference and control list	<input type="checkbox"/>	<input type="checkbox"/>
c Transaction document identifier	<input type="checkbox"/>	<input type="checkbox"/>
d Transaction identifier assigned at terminal entry	<input type="checkbox"/>	<input type="checkbox"/>
2 Are multiple transaction references stored in a ledger account record?		
a Record of every transaction since the last annual closing	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
_____		
b Record of transactions since last listing of transactions by account	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
_____		
c Reference chain	<input type="checkbox"/>	<input type="checkbox"/>
Describe _____		
_____		



---

# Audit Software Selection Checklist



## Hardware/Software Requirements

	Required by or Available to Auditor	Required by or Used by Package
1 Computer		
Manufacturer	_____	_____
Model	_____	_____
Primary storage available for package	_____	_____
2 Operating system	_____	_____
3 Peripheral devices		
Input	_____	_____
Output	_____	_____
Secondary storage	_____	_____
4 Other hardware/software requirements (specify)	_____	_____
	_____	_____
	_____	_____
5 Portability of package among installations and hardware (specify)	_____	_____
	_____	_____
	_____	_____

**Capabilities, Features, and Functions**

	Required by Auditor	Available With Package
1 Data access capabilities		
a Media		
<i>Tape</i>	_____	_____
<i>Disk</i>	_____	_____
<i>Other (specify)</i>	_____	_____
b File organization/structure		
<i>Sequential</i>	_____	_____
<i>Index sequential</i>	_____	_____
<i>Direct access</i>	_____	_____
<i>Databases (specify)</i>	_____	_____
c Record type		
<i>Fixed-length records</i>	_____	_____
<i>Variable-length records</i>	_____	_____
<i>Undefined records</i>	_____	_____
d Data format		
<i>Zoned decimal</i>	_____	_____
<i>Packed decimal</i>	_____	_____
<i>Binary</i>	_____	_____
<i>Alphanumeric</i>	_____	_____
2 Selection and statistical functions		
a Selection capabilities		
<i>Based on criteria supplied by auditor</i>	_____	_____
<i>Specification of maximum number to be selected</i>	_____	_____

Required by Auditor	Available With Package
---------------------	------------------------

b Sampling capabilities

<i>Every n<sup>th</sup> item (interval or systematic sampling)</i>		
--	--	--

<i>Random systematic sampling (specify number of random starts)</i>		
---	--	--

<i>Estimation sampling</i>		
----------------------------	--	--

<i>Discovery sampling</i>		
---------------------------	--	--

<i>Sampling where probability is proportional to size</i>		
---	--	--

c Population and sampling population analysis

<i>Stratification of population</i>		
-------------------------------------	--	--

<i>Calculation of sample size</i>		
-----------------------------------	--	--

<i>Analysis of sample results</i>		
-----------------------------------	--	--

<i>Calculation of means and standard deviation</i>		
--	--	--

d Random number generation

<i>Random number generation</i>		
---------------------------------	--	--

<i>Same seed for random number generation yields same sequence (useful in testing)</i>		
--	--	--

3 Computation capabilities

a Arithmetic

<i>Addition, subtraction, multiplication, and division</i>		
--	--	--

<i>Maximum calculations per field (specify)</i>		
---	--	--

<i>Maximum number of calculations per run (specify)</i>		
---	--	--

<i>Restrictions (specify)</i>		
-------------------------------	--	--

	Required by Auditor	Available With Package
<i>b</i> Automatic totaling and subtotaling		
<i>Maximum number of fields (specify)</i>	_____	_____
<i>Maximum number of subfield levels (specify)</i>	_____	_____
4 Comparison capabilities		
<i>a</i> Comparison between		
<i>Two data items in same record</i>	_____	_____
<i>Data items with value supplied by auditor</i>	_____	_____
<i>Data items with value computed by package</i>	_____	_____
<i>Data items on two different files</i>	_____	_____
<i>Data items on more than two files</i>	_____	_____
<i>Data items on files with different formats</i>	_____	_____
<i>b</i> Logic functions		
EQUAL NOT EQUAL	_____	_____
LESS THAN	_____	_____
GREATER THAN	_____	_____
AND	_____	_____
OR	_____	_____
NOT	_____	_____
5 Record handling capabilities		
Sorting	_____	_____
Merging or combining data from two or more files	_____	_____



	Required by Auditor	Available With Package
6 Output capabilities		
Maximum number of reports generated in a single run	_____	_____
Automatic formatting of report	_____	_____
Auditor-specified report format	_____	_____
Bar graphs	_____	_____
Confirmation request output	_____	_____
Produce new auditor-selected data file	_____	_____
Update, add, and delete records in an auditor-selected file	_____	_____
7 Other functions and capabilities		
Table reference function	_____	_____
Generation of operating system control statements	_____	_____
Exits to allow user-provided routines (specify)	_____	_____
Suitable handling of invalid output data (specify)	_____	_____

**Usage Considerations**

	Required by or Available to Auditor	Required by or Provided by Package/Vendor
1 Expertise required of field auditor using package		
General EDP expertise	_____	_____
Package-specific expertise	_____	_____
<i>Coding</i>	_____	_____
<i>Checklist</i>	_____	_____

400 Appendix E

	Required by or Available to Auditor	Required by or Provided by Package/Vendor
2 Documentation		
Assessment of documentation quality needed/available in terms of coverage and level of explanation	_____	_____
Auditor-oriented documentation	_____	_____
3 Days of training needed/provided for -		
Field staff	_____	_____
Package specialists on audit firm staff	_____	_____
4 Package expertise need and availability		
Vendor consultation service (specify)	_____	_____
On-site assistance (specify)	_____	_____
5 User-oriented capabilities		
Editing of syntax of auditor instructions	_____	_____
Audit trail of use suitable for working papers	_____	_____

**Vendor Support**

- 1 How much effort has the vendor been expending to keep the software up to date? How often are updates issued?
- 2 Are updates automatically sent to the users of the package? To both the purchasers and the lessees?
- 3 Is the vendor a reliable financially stable organization? How many years has it been in business?
- 4 How many organizations currently use the package?
- 5 Will the vendor give you the names of some users as a reference?
- 6 Is there a user group established to share information?

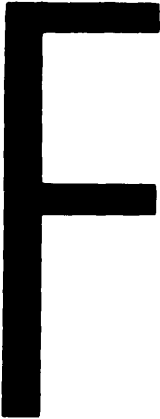
**Costs**

- 1 What is the cost of acquiring the package? Leasing? Purchasing?
- 2 What is the cost of updates to the package?
- 3 What is the cost of the consultation service?
- 4 What is the charge for training? For how many persons? Charge for additional trainees?
- 5 What is the cost of the forms and supplies? Checklists or coding forms? Confirmation forms? Output tapes or disks?
- 6 How efficient is the package for your type of applications? Will the vendor allow you to run a benchmark test?
- 7 What is the cost of machine time? Will there be service center or time-sharing costs?



---

# Evaluation of Controls Over Processing Performed by a Data Processing Service Center



**Background**

Name of data processing service organization

\_\_\_\_\_

Address \_\_\_\_\_

\_\_\_\_\_

Name and title of person at service center responsible for client's jobs \_\_\_\_\_

\_\_\_\_\_

Major computer equipment available at service center \_\_\_\_\_

Client applications being processed by service center

Application

Frequency

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Name and title of person in client organization responsible for liaison with service center.

\_\_\_\_\_

**Use of Third-Party Review Report**

Yes      No

- 1 Is a third-party review available?       
  - a Type of report \_\_\_\_\_
    - Report on the design of the system
    - Report on both the design of the system and compliance tests
  - b Date of the report \_\_\_\_\_
- 2 Is the third-party review adequate for reliance?       
  - a Service auditor \_\_\_\_\_
    - Is professional reputation satisfactory?
    - b Are the control procedures described in the report adequate?
    - c Is the scope of the compliance tests adequate?
    - d Is the timing of the report adequate?

**General Review of Data Processing Service Center Operations**

Yes      No

- 1 Have provisions been made by the service center for backup in case of equipment failure?
- 2 Do service center personnel have access to client assets (e.g., charge accounts and bank accounts)?
- 3 Does security over client records at the service center appear adequate?
- 4 Does the service center or the client have adequate insurance to protect against substantial loss (lost data, lost files, or errors in processing)?
- 5 Does the client's agreement with the service center provide for client ownership of client files maintained by the service center?

**Control Over Input**

- 1 Are the originals, or copies, of all source documents transmitted to the service center retained for a reasonable period of time?
- 2 Is a control established for all data sent for processing?       
  - Document numbering
  - Document count
  - Transaction count
  - Control totals (list below)

---



---

- |   |                          |                          |
|---|--------------------------|--------------------------|
| 3 Do input and validation controls in the program appear adequate to detect incorrect input data?                   | <input type="checkbox"/> | <input type="checkbox"/> |
| 4 Does the client reconcile independent input control figures with control figures furnished by the service center? | <input type="checkbox"/> | <input type="checkbox"/> |

**Control Over Master File Changes**

- |   |                          |                          |
|---|--------------------------|--------------------------|
| 1 Is a control printout provided for the client for all master file changes?  | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 Is there a master file control record count or other control total provided for the client to check for the loss or nonprocessing of master file records? | <input type="checkbox"/> | <input type="checkbox"/> |
| 3 Does client check and reconcile control figures contained in item 2?  | <input type="checkbox"/> | <input type="checkbox"/> |
| 4 Are master file changes reviewed by someone other than their originator?  | <input type="checkbox"/> | <input type="checkbox"/> |

**Control Over Error Corrections and Resubmissions**

Yes                      No

- |   |                          |                          |
|---|--------------------------|--------------------------|
| 1 Is there an adequate error printout identifying all errors?   | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 Does the client have adequate procedures for recording errors and controlling correction and resubmission of input? | <input type="checkbox"/> | <input type="checkbox"/> |
| 3 Do error procedures of both the service center and the client appear adequate for handling—                         |                          |                          |
| a Unmatched transactions?   | <input type="checkbox"/> | <input type="checkbox"/> |
| b Control total or control count differences?   | <input type="checkbox"/> | <input type="checkbox"/> |
| 4 Is there a follow-up to ensure that corrections are properly made?  | <input type="checkbox"/> | <input type="checkbox"/> |

**Control Over Output**

- |  |                          |                          |
|--|--------------------------|--------------------------|
| 1 Are client procedures adequate for reviewing or testing output prior to distribution (including recalculation for test items)? | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 Does the client have adequate procedures for controlling the distribution of output?   | <input type="checkbox"/> | <input type="checkbox"/> |

**Adequacy of Audit (Management) Trail**

- |   |                          |                          |
|---|--------------------------|--------------------------|
| 1 Is there a listing of input data or some other satisfactory means for identifying transactions processed by the service center? | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 Is there a periodic printout of ledger balances or a satisfactory alternative for providing this information?                   | <input type="checkbox"/> | <input type="checkbox"/> |
| 3 Is the audit (management) trail complete and adequate for all significant transactions?   | <input type="checkbox"/> | <input type="checkbox"/> |

**Adequacy of Protection and Security**

- |   |                          |                          |
|---|--------------------------|--------------------------|
| 1 Does the service center appear to have adequate provisions for file reconstruction? | <input type="checkbox"/> | <input type="checkbox"/> |
|---|--------------------------|--------------------------|

406 Appendix F

- 2 Does the client have provisions for "worst case" file reconstruction (that is, if all files at the service center are destroyed)?
- 3 Does the client organization have complete documentation and a copy of the source coding for all programs written specifically for it and for which full payment has been made (client ownership)?



---

# Index

- abbreviations, standards for, 48
- access control, 8, 14, 17, 108, 113, 324–327
  - computer room, 119–120, 324–327
  - data, 116–118
  - database, 179
  - documentation, 17, 120, 324
  - terminals, 14, 120, 325–327
- accounting controls
  - computer programs, 131
  - definition, 11
  - management practices, 55–57
- accounting information for EDP system, 102–103
- accounting practice research, 5
- accounting services and the computer, 5
- accounts receivable, case example, 214
- address checking, disk pack, 93
- adjustment filter, in error correction, 117
- administrative control, 11, 15, 55–56
  - for computer programs, 131
- aging of accounts receivable audit trail example, 195
- AICPA Code of Professional Ethics, 10
- American National Standards Institute, 48
- analyst/programmer, 40
- anticipation control, 340
- application
  - definition, 6, 133
  - list, 316–317
  - postimplementation review, 8
  - risk and exposure analysis, 20
  - system development and maintenance controls, 17, 50–51, 320
- application controls, 7, 12, 18, 30, 133, 299, 334–344
- application development life cycle, 17, 51–53, 57
  - alternatives to, 53
- applications programmer, 39
- approval, 154, 187
- aptitude tests, use of in personnel selection, 47
- assignments, rotation of, 47
- audit hooks, 254, 265
- audit indicator (AI), 253, 261–262
- audit modules, 254, 263–265
- audit plan, 21–24
  - advance preparation for EDP technique use 23
  - microcomputer, 292
- audit processing facilities within applications, 263–265
- audit program, *see* audit plan
- audit software
  - customized and specialized, 220–221
  - generalized system, 23, 201–217, 222
  - selection checklist, 395–401
  - types, 207–211
- audit trail, 269, 284, 336, 338
  - checklist, 389–393
  - definition of, 183–186
  - elements of, 186–192
  - examples of, 193–196
    - for minicomputer, 291
- Auditor's Study and Evaluation of Internal Control in EDP Systems, Audit and Accounting Guide*, 17, 37, 113
- authorization, 14, 17, 153–154, 179
  - minicomputer, 294
- automatic retry and diagnosis, 86
- auxiliary storage, 140
  
- backup and recovery, 12, 15, 18, 23, 125–130, 143, 149, 178, 332–333, 344
  - database management system, 269–270
  - plan, 8, 48, 332
  - service center, 280–281
  - storage, 332
- badges, 119
- base of check digit, 156
- batch,
  - controls, 6
  - control totals, 130–139, *see* control totals
  - data preparation and input, 168–171
  - entry/batch processing method, 148–149
  - lists, 187
  - reference, 190
  - size of, 168
- batch ticket, 171
- baud, 93
- before and after images, 177, 190, 269
- before and after values, 172
- bonding of data processing personnel, 111, 115
- breakpoints, 178
- byte, 85

- cancellation of source documents 169
- card reader hardware controls 87
- central processor hardware controls 85-86
- chaining 270
- change notice program 44 62-63 123-124
- check digits 95 139 156-157 338
- chief programmer 51
- circulating error file 162-164
- class identification, 176
- classification of controls 11-13
- COBOL, 175 178 243
  - example 380-383
- code
  - error correction 44-85 90-91
  - Hamming, 85 90 93
  - logic tracing for test data 225
  - review 240-244
  - valid transaction 155
- common body of knowledge for auditors 305
- comparison
  - by audit software 205
  - of recorded accountability with assets 14
  - of stored data records with assets 14 180
  - with approximations 180
- completion phase of the review 7 27 31
- computer audit specialists, 9, 304 306-307
- computer continuation provisions see continuation controls and provisions
- computer facilities
  - access 119-120 325-327
  - questionnaire 314
- computer operator 40
- computer room
  - access, 8 14 17 119-120 324-327
  - location and construction 114
- computer service organizations, 279-281
  - Audit and Control Considerations in a Minicomputer Environment* Computer Services Guideline 293
- computer system alternatives to manual controls 6-7
- concurrent
  - processing 100
  - testing 253
- contingency plan 332
- continuation controls and provisions 8, 12, 15-16, 331-333, 344
  - minicomputer 292
- control clerk, 6, 40 42 45
- control flowchart 364
- control function 18 315
  - database 110-111
  - duties of, 45-47 327-328
  - minicomputer 297
- control
  - access, see access control
  - error correction, see error correction
  - input see input control
  - output, see output control
- controls
  - applications see application controls
  - continuation, see continuation controls and provisions
  - classification of 11-12
  - development, 12 320
  - general see general controls
  - hardware see hardware controls
  - controls (*cont*)
    - service center 280-281 403-405
    - to prevent or detect error 167-168
    - to protect against loss of data 113
  - control totals 124 157-158 171 177 340
    - of program code 245
    - verification of 138-139 327
  - conversion 138 169
  - correction screening 177
  - corrections, handling of, 161-162
  - corrective controls 12
  - corroborative inquiry 32
  - criteria selection by audit software 202
  - cross-reference systems, 72 103
    - in code review 244
  - crossfooting test 176-177
  - customized audit reports 273-274
  - customized audit software 220-221
  - cycle data processing 135
- data administration function 6 17 56
- data
  - capture 135-137
  - conversion 138 169
  - entry 135-137 169-173
  - preparation 6 138-139 168 335
  - review 168
  - safeguarding access and use 116-118
  - validation see validation
- data collection devices, 171
- data communication 3, 93-95
  - distortion in 94
- data controls service center 281
- data description language database, 106
- data dictionary 64 68 106 267 268
- data element description in documentation 66
- data entry operator, 40
- data manipulation language, 106
- data processing
  - cycle 135-140
  - liability insurance 115-116 331
  - review of management 8
  - system 5 29
- data set 94
- database, 4 105 141
  - access controls 179
  - changes 190
  - control concerns 108-109
  - controls 329
- database administrator 40 44 46 109-111, 179 267 329
- database management systems 99-111, 141
  - audit procedures 274-275
- debugging standards 49
- default option 174
- delay method for corrections 161
- derived transactions 189
- "design of the system" service auditor report 283
- design phase review 9
- desk checking 49
- detective controls 12
- development controls, see controls, development
- development life cycle see application
  - development life cycle
- development support library 51

- direct access file organization, 140–141
- direct entry, 171–173, 337
- disasters, 113
  - contingency plan, 8, 48, 332
- disk files, retention, 129–130
- disk storage hardware controls 91–93
- distributed data network, 141
- distributed data processing, 4, 290
- distribution control for output, *see* output control
- document counts 158
- document identifier, 190
- documentation
  - access control, 17, 120, 324
  - aids, 69–77
  - application, 59–82
  - database, 108
  - example, 367–388
  - minicomputer, 296–297
  - minimum, 75–79
  - standards, 321–322
- dual read control, 87
- dump utility, 248
- dump/restore, database 106
- dumps 129
- duplicate process check, 84
- dynamic storage allocation 100
- dynamic testing, 253
  
- echo check, 84, 95, 157, 338
- editing, *see* validation
- EDP audit specialists, 9, 304, 306–307
- EDP functions, 13
- Effects of EDP on the Auditor's Study and Evaluation of Internal Control* 7
- electronic components failure of, 83
- emergencies, 48
- encryption, 123, 125
- entry *see* direct entry
- equipment
  - checks and controls 85
  - malfunctions 113
- error
  - prevention and detection, 167–168 180
  - rate of, 95 169–170
  - sources of 167–168 173–175
- error control, 18 328
  - data communication, 93–95
  - data preparation, input or entry 167–168
  - processing, 32, 176–179
  - report 368–371
- error correction code 84–85 90–91
- error correction 284
  - control over, 161–164
  - procedures 67 95
  - screening 177
  - retry 85 89 93
- error file 139 162–164
- error flag 162
- error handling 161, 339
  - by operating system 101
- error message list 388
- ethics, 10
- executive *see* operating system
- exits, *see* program exits
- extended records, 254 262
  
- external label 117
- external review, 319
  
- facilities management, 279
- facilities, safeguarding access and use of, 113–131
- fading, in data communications, 94
- failure logs, 96
- file
  - classification, 140–141
  - conversion, minicomputer, 295
  - description generators, 72–73
  - direct access, 140–141
  - documentation, 61, 64–67, 385–386
  - labels, 102, 117–118, 278, 323, 328
  - organization for processing 140–141
  - physical safeguards, 116–117
  - protection ring, 118
  - quality maintenance, 164
  - retention, 48
  - review, 164
  - testing 220
  - updating, 140
  - usage controls, 178
- file-order batch processing, 148, 150–152
- fire prevention, 114
- extinguisher, 115
- flowchart
  - control, 364
  - horizontal, 365
  - logic tracing for test data, 225
  - packages, 69–72
  - program, 62, 345, 372–379
  - run-to-run, 61
  - system, 345, 363, 365
- flowcharting, 345–365
  - software, 103
  - software, in code review, 224
- formatting program, 72–76
- fourth generation languages, 221
- full duplex mode, 95
  
- general access controls, 113
- general controls, 7, 12, 16–18, 30
  - minicomputer, 293–298
  - questionnaire, 318–338
- generalized audit software, 23, 201–217, 222
  - selection checklist, 389–393
  - types 201–211
  - within a database management system 273–274
- generalized input data validation 158–161
- glossary in documentation, 48
- graph analysis program, 247
  
- Halon gas fire extinguisher, 115
- Hamming code 85, 90 93
- hardware configuration questionnaire 314
- hardware controls 17 83–97 322
  - and the audit 96
- hardware maintenance 48
- hash totals, 124 158
  - of program code, 245
- head crash 91

## 410 Index

- header label, 117
- heuristic application development, 53
- hierarchical index, data dictionary, 68
- high-level languages, 221
- hole count control, 87
- horizontal flowchart, 365
- housekeeping, 48, 103, 115–116
- human errors, 113
- humidity control, 116
  
- IF statement, in testing, 225
- index method of file organization, 141
- indexed sequential file organization, 141
- indexes, data dictionary, 68
- industry software, 222–224
- information system master development plan 6, 50
- initiation of transactions*, 335
- input control, 6, 18, 139, 171–173, 339, 341
- input data validation, *see* validation
- input editing, *see* validation
- input/output chart 67
- inquiry software 23
- instruction count, program, 245
- insurance, 115, 116, 331
- integrated test facility (ITF), 253–256
- intelligent terminals, 3, 139
- internal auditors, 9, 298
- internal control, 11, 13–15, 27, 30
- internal file label, 102, 117, 178, 323
- IRS Revenue Procedure 64-12, 75–77, 80–81
- IRS Revenue Ruling 71-20, 75–77, 82
- interpreter, for audit software, 207
- invalid values test, 155
- ITF, *see* integrated test facility
  
- job, 60
  - accounting facilities, 236–242
  - control language, 49, 67
  - descriptions, 39–40
  - documentation, 60, 67
  
- key entry, 169–170
  - verification, 170
- keydisk, 138–139, 169
- keypunch, 138, 169
- Keyword in Context (KWIC) index in data dictionary 68
  
- labels, 102, 117–118, 178, 323, 328
- last-time-through logic, 175, 225
- lateral check 90
- leased lines in data communications, 94
- librarian
  - function, 6, 17, 40–46, 56, 120–121 324
  - software 75–77 125, 246
- limit and reasonableness test, *see* range and reasonableness test
- line and equipment diagnostics, 95
- linkage and chaining reports for database, 270
- linkage editing, 102
- list organization 141
  
- log
  - analysis software, 104
  - database management systems, 110, 269
  - input and transaction, 144–145, 338
  - minicomputer, 291
- logging facilities, 235–236
- logical batches, 172
- logical lists, 187
- logic tests, 155–156
- longitudinal parity, 90, 93
  
- machine operation, 48
- magnetic ink character reader, hardware
  - controls, 95
- magnetic stripe cards, 119
- magnetic tape
  - file retention, 128–129
  - hardware controls 88–91
- main line module, 51
- maintenance
  - controls, 320
  - hardware, 48
- malfunction of equipment, 83
- management advisory services, 5
  - as EDP audit specialists, 304
- management and control of application systems
  - development, 17, 50–51, 320
- management of computer installations, 47–49
- management trail, *see* audit trail
- master control program, *see* operating system
- master file
  - changes, 284
  - in testing, 224
- master index, 75
- master plan, information system 6, 50
- memo updating, 143–145
- merging, 140
- MICR, 170
- microcomputer, 289, 300–301
  - audit tests, 300
- microfilm, 128
- minicomputers, 3, 222, 289–300
- missing data test, 155, 340
- modem, 94
- modular programming, 51
  - example, 380–383
- modulus, of check digit* 156
  
- NAARS (National Automated Accounting Research System), 5
- National Automated Accounting Research System (NAARS), 5
- National Bureau of Standards, algorithm for encryption, 123
- National Fire Protection Association, recommendations for computer installations 114
- network administrator, 42, 44
- noise in data communication 94
- nucleus 51
  
- OCR, 170
- odd-parity check, 85

- off-site storage, 12, 332
- on-line audit monitor, 253, 259–260
- on-line entry/batch processing method, 146
- on-line processing systems, 4, 142–145
- OPEN instructions, in code review, 117
- operating procedures, 8
- operating system, 99–102, 287
  - audit considerations, 102–103
- operation controls, 178–179, 327–329
- operations documentation, 67, 78
  - minimum for audit, 78
- operator documentation, 178
- optical character recognition equipment,
  - hardware controls, 95–96
- optimizers, 103
  - in code review, 245
  - to evaluate test data, 227, 231
- organization, EDP auditing function in a CPA firm, 303–304
- organization of EDP, 39–44
  - charts, 41–44
  - controls, 17, 318–319
  - questionnaire, 315
- output, 140
  - control, 18, 179–180, 284, 342
  - distribution, 179, 328, 343
  - log, 180
  - summary, in documentation, 61
- output review, 158–159
  - checklist, 180
  - by service center, 282
- overflow location, in random access storage, 141
- overrides, 339
  
- parallel processing to test application processing, 231–232
- parallel tests, 55
- parameters, incorrect program, 174
- parity bit check, 84–86, 88–89, 94
  - for disk, 91
  - for magnetic tape, 90, 93
- passwords, 119, 121–123, 125
- payroll processing, audit trail example, 193
- personal identification number, 119
- personnel
  - selection, 47
  - termination, 326
- pilot tests, 55
- plan of organization for EDP, 39–44
- pointers, 270
- point-of-sale devices, 171
- postimplementation review, 8–9, 52
- power supply, 96
- precision errors, 174–175
- precompilers, for audit software, 207
- preliminary phase of the review, 7, 27, 29–31, 33
- preprocessors, 72
- preventive controls, 12
- preventive maintenance, 95, 115–116
- primary storage, 140
- privacy compliance review, 9
- procedural controls, 18, 168
- processing,
  - concurrent, 100
  - processing (*cont*)
    - control functions, 45–47, 315
    - controls, 6, 12, 18, 56, 176–177, 341
    - immediate, 143, 145
    - logic, 32, 175
    - operations, 139–140
    - reference, 192
  - processing errors, 178–179
    - sources, 173–175
  - processing methods, 142–149
    - effect on audit, 135
  - processing trail, *see* audit trail
  - process tracing data, 177, 189–190
  - production programs, 124
  - program change
    - controls, 32, 321
    - detection of, 345–346
    - minicomputer, 296
    - notice, 44, 62–63, 123–124
    - using utilities, 104
  - program
    - code, review of, 240–247
    - coding conventions, 48
    - instruction hash totals, 245
    - parameters, 174
  - program documentation, 62–63, 78, 321–322, 367–388
    - minimum for audit, 78
    - narrative, 67, 322, 369–371
    - summary, 61, 322
  - program exits, 211
  - program flowchart, *see* flowchart, program
  - program storage, by operating system, 103
  - programmer, 39–40
  - programming control facility, 247
  - programs, safeguarding access and use, 113–131
  - prototyping, 53
  - pseudo-updating, 143–144
- quality assurance, in programming, 51
- quality manager, 42
- query and report processing facilities, 106, 247
- query facilities
  - audit reliance, 272
  - for testing files, 220
  
- random access, 140–141
- random number generation, 248–249, 251
- range and reasonableness test, 139, 155, 176, 338, 340
- read-only
  - notch for diskettes, 118
  - switch for disks, 118
- record counts, 158
- record layout software, 103
  - in code review, 245
- record retention, 82
- recovery, 126–127, 172, 178
- REDEFINES clause, in code review, 243
- redundant character check, 84, 156
- reference and control list, 187–189
- reference data, 142, 146–148
- reject rate, 95

## 412 Index

- release form, 180
- reorganization, database 106
- rerun breakpoints, 178
- rescan 95
- resource allocation by operating system 100-101
- restart and recovery instructions, 67 235
  - example, 388
- retention procedures, *see* backup and recovery
- retry for errors, 85, 89, 93
- Revenue Procedure 64-12, 75-77, 80
- Revenue Ruling 71-20, 75-77, 82
- review
  - application postimplementation 8-9, 52
  - computer processing continuation provisions, 8, 34-35
  - data processing management, 8
  - design phase, 9
  - external, 319
  - physical disaster protection, 35
  - privacy compliance, 9
  - program code, 240-244
  - system operations, 235-240
- risk and exposure analysis, 19-21
- rotation of assignments 47
- ROUNDED instruction, in COBOL, 175
- rounding error 174-175
- route slips 169
- row check 90
- Rule 201, Code of Professional Ethics, *see* AICPA Code of Professional Ethics
- run book, 179
- run-to-run
  - controls, 179
  - flowchart 61
- safeguarding
  - computer facilities, 114-120
  - computer programs and data, 120-126
- sales order entry, audit trail example, 193
- sample output, 62
- sampling with audit software, 202
- SAS 1 *Codification*
  - section 150 02, 27
  - section 320 (*see also* SAS 3), 13
- SAS 3 *The Effects of EDP on the Auditor's Study and Evaluation of Internal Control*, 7 31 282
- SAS 30 *Reporting on Internal Control* 19
- SAS 39 *Audit Sampling on risk of material error* 19
- SAS 44 *Special-Purpose Reports on Internal Accounting Control at Service Organizations* 283-285
- schedulers 42 44
- scheduling 48 100
  - audit activities, 22-24
- screen formats, 168
- screening of corrections 177
- secondary (auxiliary) storage 140
- security 284
  - codes 17 326
  - facility 123
  - for minicomputers, 298-299
  - officer 42 44
  - security (*cont*)
    - physical 18, 48
    - service center 281
    - software, 125
- segment description in documentation, 65
- segregation of functions, 13, 17, 40-41, 67, 318
  - database administration, 110
  - EDP and users 41, 56
  - minicomputer, 291, 293-294
  - within EDP, 56
- SELECT clause in code review, 243-244
- selection functions of audit software, 202-203
- self-checking number, 156
- self-diagnosis of equipment, 86
- sequence test, 155, 176, 338, 340
- sequential file organization, 141, 149
- sequential processing, 140, 148
- service auditor, 283, 285
- service center, 279-281
  - auditor concerns 282-283
  - evaluation, 403-405
- signature verification systems 120
- simulation to test application processing, 231-232
- simulators, 103
- small business computer systems 222
- SMF (System Management Facility) 236
- snapshot technique, 254, 262-263
- software
  - audit *see* audit software
  - catalogs 248
  - documentation aids, 69-75
  - industry, 222-224
  - review aids, 244
  - questionnaire, 315
- sorting, 140, 152
- source code
  - compare, 104 247
  - listing 62
- source documents
  - retention, 127-128
  - summary, in documentation, 61
- source records in audit trail, 186
- special journals, 187, 192
- spooled output 140
- staff training, 308-309
- standard operating procedures 56
- standards
  - abbreviations 48
  - debugging, 49
  - documentation 321-322
  - EDP 48-49 243
  - minicomputers 291
  - technical proficiency 305
- static system test 224
- statistical analysis package, 248
- statistical functions of audit software 202-225
- storage 85-86
  - devices 140
  - protection 87 100
- stored data
  - comparison 157 338
  - controls 342
- string tests 54
- structured programming 51
- study and evaluation of internal control in EDP systems 22 27-35

- substantive testing 22
  - minicomputer, 300
- supervisor, *see* operating system
- supervisory terminals, 172
- SUPERZAP utility software, 104
- suspense file, 162
- system, *see also* application, 133
  - activity logs, 235
  - data processing, 5, 29
  - design, trends, 4
  - documentation, 60–61, 77–78
  - file testing, 222
  - management, 86, 96
  - tests, 54, 295
- system development life cycle, *see* application development life cycle
- system flowchart, 345
- system generation (SYSGEN), 100
- system software, 5, 17, 99–112
  - controls, 323
- systems analyst, 39
- systems and procedures manual, 47–49
- systems control audit review file (SCARF), 253, 256–259
- systems programmer, 39
- SYSGEN (system generation), 100
  
- table values, internal, 174
- tagging, 253
- tape cleaner, 89
- tape unit monitoring, 91
- tax practice and the computer, 4–5
- technical proficiency for auditing
  - computer-processed records, 9, 305–307
- templates, on keys, 168
- terminal entry, 14, 171–173, 337
  - data, 137
- terminal entry/batch processing, 146–148
- terminal entry/on-line processing, 142–145
- terminal identifier, 187
- terminals, access, 14, 120, 325–327
- terminated employees, 326
- test data, 224–225, 254–256
  - generator, 103, 227–229, 247
- test deck, *see* test data
- test programs, 124
- test results, 62
- testing and debugging aids database management systems 270–272
- testing
  - auditor's role, 55
  - concurrent, 253
  - dynamic, 248
  - of applications 54–55, 57
  - of programs 57
- tests of compliance, 31
- tests of controls, 27, 31–32
- third-party reviews
  - code, 246–247
  - evaluation checklist, 403–405
  - service center, 283–284
- third standard of fieldwork, 11
  
- threats, 113
- time-sharing services, 248–249, 286–288
  - control features, 287–288
- timing
  - of computer-related audit activities, 22
  - of error corrections, 161
- totals, *see* control totals
- trace facilities, database management systems, 272
- tracing data, 177
- tracing, in audit trail, 189
- track parity, 90
- trailer label, 117
- training, in audit software, 213
- transaction
  - audit log, 236
  - code, 155
  - date filter, 177
  - derived, 189
  - log, 172
  - risk and exposure analysis, 20–21
  - selection methods, 256–260
  - source record in audit trail, 186–187
  - source reference, 190–192
- transaction order batch processing, 148
- transcription error, 156
- transmittal controls, 169, 180
- transposition error, 156
- triggered transactions, 189
- turnaround document, 137, 171
- turnkey systems, 296
  
- unauthorized or undocumented changes, *see* program changes
- user auditor, 283, 285
- user controls, 56, 284–285
  - service center, 281–282
- user documentation, 67–69
- user-developed systems, 53
- utilities, database management, 106
- utility software, 5, 103–104, 221–222, 248
  
- validation, 6, 12, 139, 147–148, 155, 169–170, 172, 338–340
- validity check on equipment, 87
- validity of corrections, 161
- vendor libraries, 247–248
- verification, 6, 138, 168, 170
  - of control totals, 138–139, 327
- version of program, 173
- visual scan, 180
- voice verification systems, 120
- volume test, 55
  
- walk through, 51
- weights for check digit, 156
- write-enable ring, 118





M013509