System/36

Work Station Utility Guide

When You Are:	You Can Find Information In:
Planning to Install Your Computer	What to Do Before Your Computer Arrives or Converting from System/34 to System/36
Getting Your Computer Ready to Use	Setting Up Your Computer Performing the First System Configuration For Your System System Security Guide
Operating Your Computer	Learning About Your Computer Operating Your Computer
Operating and Using the Utilities	Source Entry Utility Guide Data File Utility Guide Creating Displays Work Station Utility Guide Utilities Messages
Programming Your Computer	Concepts and Programmer's Guide System Reference Sort Guide **Work Station Utility Guide** (language manuals) (language message manuals)
Communicating with Another Computer or Remote Device	(communication manuals) (communication message manuals)
Determining the Cause of a Problem	System Messages (message manuals) System Problem Determination

IBM System/36:
Work Station Utility Guide

SC21-7905-2

What Is Your Opinion of This Manual?

Your comments can help us produce better manuals. Please take a few minutes to evaluate this manual as soon as you become familiar with it. Circle Y (Yes) or N (No) for each question that applies. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

FINDING INFORMATION

Y N Is the table of contents helpful?
What would make it more helpful?

Y N Is the index complete?
List specific terms that are missing.

Y N Are the chapter titles and other headings meaningful?
What would make them more meaningful?

Y N Is information organized appropriately?
What would improve the organization?

Y N Does the manual refer you to the appropriate places for more information?
List specific references that are wrong or missing.

UNDERSTANDING INFORMATION

Y N Is the purpose of this manual clear?
What would make it clearer?

Y N Is the information explained clearly?
Which topics are unclear?

Y N Are the examples clear?
Which examples are unclear?

Y N Are examples provided where they are needed?
Where should examples be added or deleted?

Y N Are terms defined clearly?
Which terms are unclear?

Y N Are terms used consistently?
Which terms are inconsistent?

Y N Are too many abbreviations and acronyms used?
Which ones are not understandable?

Y N Are the illustrations clear?
Which ones are unclear?

USING INFORMATION

Y N Does the information apply to your situation?
Which topics do not apply?

Y N Is the information accurate?
What information is inaccurate?

Y N Is the information complete?
What information is missing?

Y N Is only necessary information included?
What information is unnecessary?

Y N Are the examples useful models?
What would make them more useful?

Y N Is the format of the manual (shape, size, color) effective?
What would make the format more effective?

OTHER COMMENTS

Use the space below for any other opinions about this manual or about the entire set of manuals for this system.

YOUR BACKGROUND

What is your job title?

What is your primary job responsibility?

How many years have you used computers?

Which programming languages do you use?

How many times per month do you use this manual?

Your name
Company name
Street address
City, State, ZIP

No postage necessary if mailed in the U.S.A.

SC21-7905-2

Cut Along Line

Fold and tape　　　　Please do not staple　　　　Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS　　PERMIT NO. 40　　ARMONK, N. Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM CORPORATION
Information Development
Department 532
Rochester, Minnesota, U.S.A. 55901

Fold and tape　　　　Please do not staple　　　　Fold and tape

IBM

IBM System/36

Work Station Utility Guide

File Number
S36-32

Order Number
SC21-7905-2

Third Edition (April 1985)

This major revision obsoletes SC21-7905-1. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

This edition applies to Release 3, Modification Level 0, of IBM System/36 Utilities Program Product (Program 5727-UT1), and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters.

Also, this publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Canada Ltd., Information Development, Department 849, 1150 Eglinton Avenue East, Don Mills, Ontario, Canada, M3C 1H7. IBM may use whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Contents

About This Manual xi
Who should use this manual xi
How this manual is arranged xi
What you should know xi
If you need more information xii
  Coding and debugging material xiii
  Naming conventions xiii
Summary of changes xiv

Chapter 1. Introduction **1-1**
Using This Manual to Write WSU Programs 1-1
  As a Guide 1-1
    Step 1. Design 1-2
    Step 2. Code 1-2
    Step 3. Enter 1-2
    Step 4. Generate 1-2
    Step 5. Test and Debug 1-3
    Step 6. Run 1-3
  For Examples 1-3
  As a Reference 1-3
What WSU Is Designed To Do 1-4
  What WSU Is Not Designed to Do 1-6
WSU Files 1-7
  Arrays 1-7
  Master Files 1-7
  Transaction Files 1-7
What WSU Requires to Generate an Application 1-8
  The File Definition 1-8
  The WSU Source Program 1-9
  WSU Procedure Command 1-9
What WSU Produces 1-10

Chapter 2. How WSU Works **2-1**
WSU Jobs and Work Sessions 2-1
WSU Modes Of Operation 2-4
Enter Mode Program Cycle 2-4
  Enter Mode Processing Levels 2-5
  Enter Mode Display Sequences 2-7
  Primary Display Sequence 2-7
  Secondary Display Sequences 2-7
  Nonsequenced Displays 2-7
  Defining Display Sequences 2-8
Review Mode 2-9
  Review Mode Variations 2-10
    Review Only 2-10
    Review with Update 2-10
    Review with Master File Update 2-10
  Review Mode Displays 2-11
    Separate Review Mode Displays 2-11
    Combined Enter/Review Mode Displays 2-11
    Display Sequences in Review Mode 2-12
  Review Mode Functions 2-13
    Page Backward 2-13
    Page Forward 2-13
    Review by Relative Record Number 2-14
    Resume Review 2-15
WSU's Role in Review Mode 2-16
  Updating Master File Records During Review Mode 2-17
Insert Mode 2-18
  Display Sequence in Insert Mode 2-19
  Inserting a Group of Records 2-19
  WSU's Role in Insert Mode 2-20
Delete Mode 2-21
  WSU's Role in Delete Mode 2-22
Display Station Fields and Job-level Fields 2-23
  Display Station Fields 2-23
    Session-Level Fields 2-23
    Mode-Level Fields 2-23
  Job-Level Fields 2-23
Display Station Local Data Area 2-24

Chapter 3. Reserved Fields and Indicators **3-1**
Using Reserved Fields 3-1

  - Date Fields 3-2
  - *RLNO 3-3
  - *SLNO 3-3
  - *ERROR 3-4
  - *RLRU 3-5
  - *RLRN 3-5
  - *RLRR 3-5
  - *USID 3-6
  - *WSID 3-6
  - *BLANK 3-6
- Indicators 3-7
- Job, Session, and Mode Indicators 3-9
  - Job Indicators 3-9
  - Session Indicators 3-10
  - Mode Indicators 3-12
- Initial Settings of Indicators 3-14
  - AC (Accept Command Key) Indicator 3-15
  - AE (Accept Sequence Error) Indicator 3-15
  - CG (Current Group) Indicator 3-16
  - DL (Delete Mode) Indicator 3-16
  - EJ (End of Job Processing Level) Indicator 3-17
  - ES (End of Sequence Set Processing Level) Indicator 3-18
  - EW (End of Work Session Processing Level) Indicator 3-19
  - IJ (Job Initiation Processing Level) Indicator 3-19
  - IN (Insert Mode) Indicator 3-20
  - IP (Input to Process) Indicator 3-20
  - IS (Initiate Transaction Sequence) Indicator 3-20
  - IW (Work Session Initiation Processing Level) Indicator 3-21
  - JA-JN, JP-JY (Job) Indicators 3-21
  - KG-KL, KQ-KY (Command Key) Indicators 3-22
  - PG (Program Mode) Indicator 3-23
  - RC (Recovery of Work Session) Indicator 3-23
  - RP (Repeated Display) Indicator 3-24
  - RR (Review Record) Indicator 3-25
  - RS (Resume Work Session) Indicator 3-26
  - RU (Rover Update) Indicator 3-26
  - RV (Review Mode) Indicator 3-26
  - SA-SN, SP-SY (Session Level) Indicators 3-27
  - U1 through U8 (External) Indicators 3-27
  - 01 through 89 Indicators 3-28

Chapter 4. Coding Files 4-1
- WSU Files 4-1
  - Master Files 4-1
    - Sharing Master Files 4-2
  - Transaction Files 4-2
    - Writing Records to and Reading Records from the Transaction File 4-3
    - Removing Trailers and Control Records from the Transaction File 4-4
    - Recovering a Transaction File 4-4
- File Processing Procedures 4-5
  - Typical Uses of the WSU Procedures 4-5
    - Prepare a WSU Transaction File for Use by a Non-WSU Program 4-5
    - Remove Blank Records 4-6
    - Create One Record Chain from Multiple Record Chains 4-6
    - Print Relative Record Numbers 4-6
    - Create a WSU File from a Non-WSU File 4-6
    - Reorder or Exclude Record Chains 4-6
    - Concatenate Multiple Transaction Files 4-6
    - Reclaim or Remove Partially Inserted Records 4-7
    - Remove partially inserted and logically deleted records 4-7
    - Recover a WSU Transaction File 4-7
- Trailers 4-8
- File Definition Coding 4-10
  - F-Specification 4-11
  - I-Specification 4-12
    - AND/OR Relationships 4-14
    - Data Formats 4-16
  - Examples of Coding F- and I-Specifications 4-21
- File Coding 4-25
  - J-Specification 4-26
  - T-Specification 4-27
  - M-Specification 4-29

Chapter 5. Coding Arrays 5-1
- Using Arrays 5-2
- Defining Arrays 5-5
  - Naming Arrays 5-6
  - Spacing Arrays 5-6
  - Ordering Arrays 5-6

- Referencing Entire Arrays For Processing 5-7
  - Processing from Array-to-Array 5-7
    - Processing Arrays of Different Lengths 5-10
    - Processing an Array with a Single Value 5-11
  - Adding All Elements within an Array 5-14
- Referencing Individual Array Elements in Processing 5-15
  - Indexing an Array 5-15
    - Specifying a Constant Index 5-16
    - Specifying a Variable Index 5-18
- Referencing Only Part of a Field 5-19
- Searching an Array for a Specific Element 5-21
  - Sorting Array Elements before a Search 5-22
  - Starting the Search at a Particular Element 5-23
  - Determining If a Search is Successful 5-25
  - Referencing an Element that Satisfies a Search 5-25
  - Searching an Array for More than One Element 5-27
  - Recording the Results of a Search 5-29
- Displaying Entire Arrays 5-30
- Displaying Individual Elements of an Array 5-31
- Loading Arrays 5-32
  - Reading Input File Data into an Array 5-32
    - Reading Data from Consecutive Positions 5-34
    - Reading Data from Scattered Positions 5-36
- Summary 5-40

Chapter 6. Coding Displays 6-1
- Guidelines for Planning a Display 6-1
- S-(Display Control) Specification 6-3
  - Identification 6-3
  - Attributes 6-4
  - Function/Command Keys and Key Mask 6-4
  - Display Sequences 6-5
  - Priority 6-7
  - Preprocess 6-7
  - Record Identifying Indicators 6-8
- D-(Field Definition) Specification 6-9
  - Data Fields 6-9
    - Input Fields 6-9
    - Output Fields 6-9
    - Output/Input Fields 6-9
  - Constant Data 6-10
    - Prompts 6-10
    - Constant Values 6-10
  - Required Entries 6-10
  - Input and Output Fields 6-11
  - Constants 6-14
  - WSU Arrays 6-15
  - Self-Check 6-15

Chapter 7. Coding Processing 7-1
- C-Specification Entries 7-2
  - Processing Function 7-3
  - Conditions 7-3
  - Operation Codes 7-4
  - Results 7-5
  - Comments 7-6
- Subroutines 7-7
- Preprocessing 7-9
- Operations 7-11
  - Arithmetic Operations 7-11
  - Array Operations 7-13
  - Branching Operations 7-19
  - Comparing and Testing Operations 7-20
    - Comparing Factor 1 to a Field or Literal 7-20
    - Comparing Factor 1 to a Table 7-22
  - Range Operation 7-24
  - Debugging Operation 7-24
  - Indicator Operations (Setting On and Off) 7-25
  - Input and Output Operations 7-25
  - Move Operations 7-30
  - Subroutine Operations 7-33
- Using PUTS in Special Processing Level Calculations 7-34
  - Examples Using PUTS 7-35

Chapter 8. Entering and Generating a WSU Program 8-1
- Entering The Code 8-1
  - Entering the File Definition 8-2
    - Storing the File Definition 8-4
  - Entering the Source Program 8-5

Storing the Source Program 8-8
Generating a WSU Program 8-9
Using the WSU Prompt Display 8-9
Name of Source Program 8-10
Name of Library 8-10
Stop Option 8-11
Source Print Option 8-11
Size of Work File in Blocks 8-12
Generation Option 8-12
Cross-Reference Level 8-13
Save Display Format Source 8-14
Cmd 4-Put on JOBQ 8-14
WSU Procedure Command 8-15
The WSU Generation Process 8-16
Generation Output 8-17
WSU-Generated Procedure 8-18
Modifying the WSU-Generated Procedure 8-19
Maximum Number of Display Stations 8-19
Message Member Name and Library 8-20
History File Logging 8-20
Never-Ending Program 8-21
Region Size 8-21
File Disposition 8-22
Debugging Option 8-22
Universal Work Session Selection Authorization 8-22

Chapter 9. Testing and Debugging WSU Programs 9-1
Diagnostic Checklist 9-2
For Errors in Generation 9-2
For Errors in Running the Program 9-2
Supporting Documentation 9-3
Diagnostic Tools 9-3
Debugging Template and Reference Manual 9-3
Generation Printout 9-4
Heading Information 9-4
Work Station Utility Options 9-5
Source Program Information 9-6
Diagnostic Information 9-8
Display Format Information 9-13
Testing 9-14
Appropriate Test Data 9-14
Normal and Expected Data 9-15
Error Data and Unexpected Conditions 9-15
Other Testing Conditions 9-16

Chapter 10. ATTENTION: OPERATORS (Running a WSU Program) 10-1
Jobs 10-2
Work Sessions 10-2
Operating Modes 10-2
Command Keys and Function Keys 10-3
WSU Command Keys 10-4
User Command Keys 10-5
Function Keys 10-5
The WSU Display 10-6
Returning from the WSU Display 10-8
WSU Aid Display 10-8
Starting a WSU Program 10-9
Entering Data on a Display 10-10
Reviewing Records 10-12
Inserting Records 10-13
Deleting Records 10-14
Ending a Work Session 10-14

Chapter 11. WSU Example Programs 11-1
Example Program 1: Inventory Adjustments/Issues/Receipts 11-1
Running IM110W 11-1
Coding Forms 11-4
Program Generation Printout 11-17
Example Program 2: Item Master File Maintenance 11-30
Running IM300W 11-30
Coding Forms 11-33
Program Generation Printout 11-42
Example Program 3: Sales Entry 11-54
Running SE100W 11-54
Coding Forms 11-55
Program Generation Printout 11-60

Chapter 12. F-(File Description) and I-(Input) Specifications 12-1
F-Specification 12-1

- File Name (Columns 7 through 14) 12-2
- File Type (Column 15) 12-2
- Record Length (Columns 24 through 27) 12-2
- Length of Key Field (Columns 29 and 30) 12-3
- Record Address Type (Column 31) 12-3
- Type of File Organization (Column 32) 12-3
- Key Field Starting Location (Columns 35 through 38) 12-3

I-Specification 12-4
- File Name (Columns 7 through 14) 12-4
- AND/OR (Columns 14 through 16) 12-4
- Record Identifying Indicator (Columns 19 and 20) 12-5
- Record Identification Codes (Columns 21 through 41) 12-5
  - Position (Columns 21 through 24, 28 through 31, 35 through 38) 12-5
  - Not (Columns 25, 32, 39) 12-5
  - C/Z/D (Columns 26, 33, 40) 12-6
  - Character (Columns 27, 34, 41) 12-6
- P/B/L/R (Column 43) 12-7
- Field Location (Columns 44 through 51) 12-7
- Decimal Positions (Column 52) 12-8
- Field Name (Columns 53 through 58) 12-8

Chapter 13. J-(Job) Specification 13-1
- Program Name (Columns 7 through 14) 13-1
- Display Format Member Name (Columns 15 through 22) 13-1
- Message Load Member Name (Columns 23 through 30) 13-2
- Library Name (Columns 31 through 38) 13-2
- Maximum Number of Display Stations (Columns 39 and 40) 13-3
- Region Size (Columns 41 and 42) 13-3
- Date/Edit (Column 43) 13-4
- Ideographic Mode (Column 44) 13-4

Chapter 14. E-(Array) Specifications 14-1
- Array Name (Columns 27 through 32) 14-1
- Number of Entries in the Array (Columns 36 through 39) 14-1
- Length of Entry (Columns 40 through 42) 14-2
- Decimal Positions (Column 44) 14-2
- Sequence A/D (Column 45) 14-2

Chapter 15. T-(Transaction File) and M-(Master File) Specifications 15-1

T-Specification 15-1
- File Name (Columns 7 through 14) 15-1
- File Definition Location (Columns 15 through 30) 15-1
  - Library Name (Columns 15 through 22) 15-2
  - Source Member Name (Columns 23 through 30) 15-2
- Not-Found Indicator (Columns 49 and 50) 15-3
- Header Record Identifying Indicator (Columns 55 and 56) 15-3
- Number of Records (Columns 57 through 60) 15-4
- Error Indicator (Columns 61 and 62) 15-4

M-Specification 15-5
- File Name (Columns 7 through 14) 15-5
- File Definition Location (Columns 15 through 30) 15-6
  - Library Name (Columns 15 through 22) 15-6
  - Source Member Name (Columns 23 through 30) 15-6
- GET Field Names (Columns 31 through 48) 15-7
- Not-Found Indicator (Columns 49 and 50) 15-7
- Error Indicator (Columns 61 and 62) 15-7
- File Definition Type (Column 63) 15-8
- Field Level (Column 64) 15-9

Chapter 16. S-(Display Control) and D-(Field Definition) Specifications 16-1

S-Specification 16-1
- Display Format Name (Columns 7 through 14) 16-1
- Format ID (Columns 15 and 16) 16-2
- Start Line Number (Columns 17 and 18) 16-3
- Number of Lines to Clear (Columns 19 and 20) 16-3
- Lowercase (Column 21) 16-4
- Return Input (Column 22) 16-4
- Reset Keyboard (Columns 23 and 24) 16-5
- Sound Alarm (Columns 25 and 26) 16-5
- Enable Function Keys (Column 27) 16-6
- Enable Command Keys (Column 28) 16-7
- Blink Cursor (Columns 29 and 30) 16-8
- Null Fill (Columns 37 and 38) 16-9
- Enter Mode Sequence (Columns 41 through 44) 16-9
  - Start (Column 41) 16-9
  - End (Column 42) 16-9
  - Entry Required (Column 43) 16-10
  - Repeat (Column 44) 16-10
- Priority (Column 46) 16-10

  - Preprocess (Column 47) 16-11
  - Review Mode Record Identifying Indicators 16-11
  - Insert Mode Record Identifying Indicators 16-11
  - Key Mask (Columns 64 through 79) 16-12
- D-Specification 16-14
  - WSU Field Name (Columns 7 through 12) 16-14
  - WSU Arrays (Columns 13 through 18) 16-15
    - Array Elements per Row or Column (Columns 13 and 14) 16-15
    - Fill Direction (Column 15) 16-15
    - Line Number Skip Factor (Column 16) 16-15
    - Horizontal Positions Space Factor (Columns 17 and 18) 16-15
  - Starting Location (Columns 19 through 22) 16-16
    - Line Number (Columns 19 and 20) 16-16
    - Horizontal Position (Columns 21 and 22) 16-16
  - Output Data (Columns 23 and 24) 16-17
  - Edit Code (Column 25) 16-17
  - Input Allowed (Column 26) 16-17
  - Data Type (Column 27) 16-18
  - Mandatory Fill (Column 28) 16-20
  - Mandatory Entry (Column 29) 16-20
  - Self-Check (Column 30) 16-21
  - Adjust/Fill (Column 31) 16-21
  - Position Cursor (Columns 32 and 33) 16-22
  - Controlled Field Exit (Column 35) 16-22
  - Auto Record Advance (Column 36) 16-23
  - Protect Field (Columns 37 and 38) 16-23
  - High Intensity (Columns 39 and 40) 16-24
  - Blink Field (Columns 41 and 42) 16-25
  - Nondisplay (Columns 43 and 44) 16-25
  - Reverse Image (Columns 45 and 46) 16-27
  - Underline (Columns 47 and 48) 16-27
  - Column Separators (Column 49) 16-28
  - Lowercase (Column 51) 16-29
  - Constant Type and Constant Data (Columns 56 through,79) 16-30
  - Continuation (Column 80) 16-32

Chapter 17. C-(Processing) Specification 17-1
  - Processing Function (Columns 7 and 8) 17-2
  - Conditioning Indicators (Columns 9 through 17) 17-3
  - Factor 1 and Factor 2 (Columns 18 through 27 and 33 through 42) 17-5
  - Operation (Columns 28 through 32) 17-6
  - Table or Message Text (Columns 33 through 52) 17-6
  - Literal (Columns 43 through 52) 17-6
  - Result Field (Columns 43 through 53) 17-9
    - Result Field Name (Columns 43 through 48) 17-9
    - Result Field Length (Columns 49 through 51) 17-9
    - Result Field Decimal Positions (Column 52) 17-10
    - Result Field Half-Adjust (Column 53) 17-11
    - Use of Column 53 to Continue Message Text 17-11
  - Resulting Indicators (Columns 54 through 59) 17-12
    - Plus or High 1 > 2 (Columns 54 and 55) 17-12
    - Minus or Low 1 < 2 (Columns 56 and 57) 17-12
    - Zero or Equal 1=2 (Columns 58 and 59) 17-12
    - Additional Uses of Resulting Indicators 17-14
  - Comments (Columns 60 through 80) 17-14

Chapter 18. Operation Codes 18-1
  - ADD 18-1
  - BEGSR (Begin Subroutine) 18-1
  - COMP (Compare) 18-2
  - DEBUG 18-2
  - DIV (Divide) 18-4
  - ENDSR (End Subroutine) 18-4
  - EXSR (Execute Subroutine) 18-5
  - GET (Get a Master File Record) 18-6
  - GETNH (Get the Next Header Record in the Transaction File) 18-7
  - GETNR (Get the Next Record in the Transaction File) 18-8
  - GETPH (Get the Previous Header Record in the Transaction File) 18-9
  - GETPR (Get the Previous Record in the Transaction File) 18-11
  - GOTO 18-12
  - IMSG (Display an Information Message) 18-13
  - LOKUP (Look for Next Element of Array) 18-14
  - MOVE (Move) 18-16
  - MOVEA (Move Array) 18-17
  - MOVEL (Move Left) 18-18
    - If factor 2 is the same length as the result field and: 18-19
    - If factor 2 is longer than the result field and: 18-19
    - If factor 2 is shorter than the result field and: 18-19
  - MSG (Display a Diagnostic Message) 18-20

MULT (Multiply) 18-22
MVR (Move Remainder) 18-22
PRTY (Priority) 18-23
PUT (Write a Transaction File or Master File Record) 18-24
PUTN (Add to a Master File) 18-24
PUTS (Show a Display) 18-25
RANGE (Range Check) 18-27
SETOF (Set Off) 18-27
SETON (Set On) 18-28
SORTA (Sort the Elements of an Array) 18-28
SUB (Subtract) 18-29
TAG 18-29
TIME (Get the Time of Day and Date) 18-30
XFOOT (Sum the Values of an Array) 18-30
Z-ADD (Zero and Add) 18-30
Z-SUB (Zero and Subtract) 18-31

Chapter 19. File Processing Procedures 19-1
WSU Extract Procedure (WSUTXEX) 19-1
  Major Functions 19-1
  Prompt Display 19-2
  Parameters 19-2
  Command Statement Format 19-5
Examples 19-6
WSU Create Procedure (WSUTXCR) 19-7
  Major Functions 19-7
  Prompt Display 19-7
  Parameters 19-8
  Command Statement Format 19-10
  Examples 19-10
WSU Recover Procedure (WSUTXRV) 19-12
  Major Function 19-12
  Prompt Display 19-12
  Parameters 19-13
  Command Statement Format 19-14
  Examples 19-14

Chapter 20. Ideographic Considerations 20-1
Using Ideographic Characters 20-1
  Space Requirements 20-1
  Usage Restrictions 20-2
  Ideographic Coding 20-2
    Concatenation Rules 20-3

Appendix A. Using Self-Check Fields A-1

Appendix B. Additional Topics and Programming Considerations B-1
  Assigning Priority to Display Format Processing and Subroutines B-1
    Examples of Assigning Priorities B-2
  Saving Total Fields B-2
  Interactive Communications Feature (SSP-ICF) B-3
  Distributed Data Management (DDM) B-3
  Delete-Capable File Considerations B-3
  Extend File Considerations B-3
  Logical Delete Considerations B-4
User-Defined Help Displays B-5

Glossary X-1

Index X-9

About This Manual

Who should use this manual . . .

This manual explains how to use WSU, the work station utility. It is intended for people who have a basic understanding of data processing concepts and perhaps some data processing experience and who have read the *WSU Introduction* or have equivalent knowledge of WSU.

Using this manual, you should be able to:

- Design WSU programs
- Code WSU programs
- Enter and generate WSU programs
- Test and debug WSU programs
- Run WSU programs and procedures
- Follow coded WSU program examples
- Interpret generation listings.

How this manual is arranged . . .

Part 1 (Chapters 1 through 11) explains how to design, code, enter, compile, test, debug, and run a WSU program. It is intended as a guide for those programming in WSU and those running WSU programs.

Part 2 (Chapters 12 through 20) is strictly reference information, including all possible entries for each column of each WSU specification sheet. Chapter 20 discusses the use of ideographic characters in WSU programs.

What you should know . . .

You need to be familiar with the following information before reading this manual:

- You need to know how to use the controls and indicators on your display screen and how to use the keys on your keyboard, such as:
  - Cursor movement keys
  - Command keys
  - Field exit keys
  - Insert and delete keys
  - Error reset key.

This information is contained in:

- *IBM 5291 Display Station Operator's Guide*, GA21-9409
- *IBM 5292 Color Display Station Operator's Guide*, GA21-9416

- *IBM 5251 Display Station Models 1 and 11 and IBM 5292 Dual Display Station Operator's Guide,* GA21-9248

- *IBM 5251 Display Station Models 2 and 12 Operator's Guide,* GA21-9323.

- You need to know how to operate your display station to use the System/36 System Support Program (SSP):

  - Signing on and signing off the display station
  - Interacting with displays
  - Using help
  - Entering control commands and procedure commands
  - Responding to messages

  This information is contained in the manual *Operating Your Computer,* SC21-9026.

- You need to be familiar with the data processing information contained in the manual *Learning About Your Computer,* SC21-9018.

- You need to be familiar with the introduction to WSU contained in the manual *Work Station Utility (WSU) Introduction,* SC21-7904.

- You need to know how to design efficient programs by being familiar with the manual *Concepts and Programmer's Guide,* SC21-9019.

- You need to know how to design and code displays by using the manual *Creating Displays: Screen Design Aid and System Support Program,* SC21-7902.

- You need to know how to use the source entry utility (SEU) to enter your source programs and to update source members. This information is contained in the *Source Entry Utility Guide,* SC21-7901.

If you need more information . . .

You might need some or all of the following manuals for additional information while using this manual.

- *System Reference,* SC21-9020, is a reference manual for all System/36 procedures, commands, and OCL.

- *Keyboard Template,* GX21-7929, identifies the command keys used by WSU.

- *Programming with RPG II,* SC21-9006, explains how to use the RPG language.

- *System Messages,* SC21-7938, contains all messages that the System/36 displays.

- *Utilities Messages,* SC21-7939, contains all printed and displayed messages that you might get on your WSU program.

- *System Problem Determination,* SC21-7919, helps you determine whether a problem is in your own program or in the computer.

- *IBM 5292 Color Display Station Programmer's Guide to Using Color*, GA21-9413, is a reference manual describing the use of color on the System/36 displays.
- *Sort Guide*, SC21-7903, explains how to use the SSP sort utility.
- *Ideographic Sort Guide*, SC09-1054, explains how to design, code, and run sort programs and procedures that use ideographic data characters.
- *System Data Areas*, LY21-0592, explains the data areas of the system.
- *Distributed Data Management Guide*, SC21-8011, explains how to use the DDM feature.

Coding and debugging material

- *RPG Control and File Description Specifications*, GX09-1035.
- *RPG Input Specification*, GX09-1033.
- *Display Format Specifications*, GX21-9800.
- *IBM 5251 Display Station Keyboard Template Assignment Sheet and Display Screen Layout Sheet*, GX21-9271.
- *WSU Job, Array, and File Specifications*, GX21-9444.
- *WSU Processing Specifications*, GX21-7936.
- *WSU/$SFGR Debugging Template*, GX21-7926.

Naming conventions

Generally, as an application is designed, it is helpful to follow a specific naming convention for programs, displays, and menus. In this manual, the following conventions are used for program, display, and menu names:

- Program names use the format aannnW, where:
  - aa identifies the type of application:

    OE means order entry

    AR means accounts receivable

    IM means inventory management

    SE means sales entry
  - nnn is a number that identifies the type of program:

    100-199 for data entry

    200-299 for inquiry

    300-399 for file maintenance

    400-499 for file update

    500-599 for sort

    900-999 for report printing and listing

  - W identifies the name of a WSU program name.

- Display names are formed by adding a D to the end of the program name that uses the display. If the program uses more than one display, a sequence of D1, D2, and so on is used.

- Menu names use the format aannnM, where:

  - aa identifies the type of application
  - nnn is a number assigned to the menu
  - M identifies the name as a menu name.

Summary of changes

The following changes have been made for Release 3 Modification 0:

- WSU now supports non-contiguous keys for alternative indexed files.

- The maximum key length for indexed files has been increased to 99 bytes.

- The Distributed Data Management (DDM) feature to let WSU programs use remote transaction and master files has been added.

- Alternative indexes can now be created over direct files.

- Various technical and editorial changes have been made to improve the quality and usability of this manual.

Chapter 1. Introduction

WSU, the work station utility, is a part of the System/36 Utilities Program Product.

WSU is designed to help you write application programs that enter data into a single transaction file *from one or more display stations*, as well as edit and verify the data that is entered. WSU will also help you write programs that inquire into or maintain master files, again from one or more work stations.

If you have read the *WSU Introduction*, much of this first chapter is going to be familiar to you, because it is a summary of the *essentials of WSU*. The second chapter in this manual then expands on some of the topics mentioned at the end of the *WSU Introduction*.

Using This Manual to Write WSU Programs

You can use this manual in three ways:

- As a guide
- For examples
- As a reference.

As a Guide

The first part of this manual is intended as a guide in completing these steps for writing and using WSU programs:

1. Designing the program
2. Coding the program on specification sheets
3. Entering the code into the system, probably with SEU
4. Generating the program
5. Testing the program and debugging it if necessary
6. Running the program.

Step 1. Design

Designing a program means planning the output, the processing, and the input. You must decide what information you need the program to produce, what processing will produce the necessary information, and what input data is available.

Chapters 1 through 3 present some information on WSU that you will need to know in order to design a WSU program; the *Concepts and Programmer's Guide* describes in much more detail the techniques for analyzing and designing application programs.

Step 2. Code

Coding a program means writing the instructions that tell the computer what data to use, how to process it, and what to do with the results. You write these instructions as specifications:

- Job (J) specification
- Array (E) specification
- Transaction file (T) specification
- Master file (M) specification
- Display format (S) specification
- Field definition (D) specification
- Processing (C) specification.

WSU also requires a file definition, which includes the following specifications:

- File description (F) specification
- Input (I) specification.

Chapters 4 through 7 discuss considerations for coding files, arrays, displays, and processing for WSU programs.

Step 3. Enter

Entering a program means getting your written instructions (your code) into the computer. You use SEU (the source entry utility) or in some cases SDA (the screen design aid) to enter your instructions. Chapter 8 discusses entering code.

Step 4. Generate

Generating (or compiling) a program means changing your instructions (called a source program) into a form that the computer can use (called a subroutine by WSU). The generating is done by the computer; all you have to do is enter the correct command that tells the computer how to generate your source program. Chapter 8 discusses generating your program.

Step 5. Test and Debug

Testing a program means running a program with some sample data to be sure that it produces the proper results. Testing helps you find errors (bugs) in your program before you run it with your actual data.

Debugging a program means correcting errors in it. After you debug your program, it is ready to use.

Chapter 9 discusses considerations for testing and debugging your program.

Step 6. Run

Running a program means using a generated program to process some data. Once your program compiles successfully, all an operator has to do is either enter the procedure command that tells the computer to run your program or select an item from the WSU display.

Chapter 7, *Coding Processing*, includes information about what happens when a program is run; you will need this information when you code the processing for your program.

Chapter 10, *ATTENTION: OPERATORS*, describes running a WSU program; it is intended for those charged with running the actual program.

For Examples

Chapter 11 contains several programs provided as examples to illustrate WSU's abilities.

As a Reference

The second part of this manual is intended as reference material.

Chapters 12 through 17 describe the specific entries that can be made for each appropriate column of the various specification sheets. Chapter 18 describes the operation codes, and Chapter 19 the transaction file processing procedures that WSU provides. Considerations for using ideographic characters are discussed in Chapter 20.

Appendixes provide formulas and summary charts.

In addition, there is a *Glossary* that defines data processing, System/36, and WSU terms that you may not be familiar with, and, of course, an *Index* to help you locate the topic you need.

What WSU Is Designed To Do

With *master files* WSU programs can do:

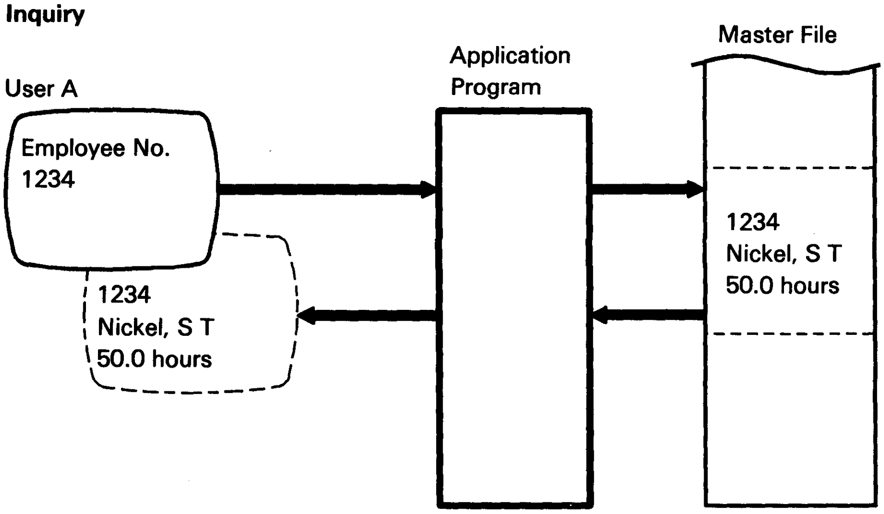
- Inquiry
- Maintenance.

With *transaction files,* WSU programs can do:

- Data entry
  - Simple
  - With edit
  - With update
- Record review
- Record insertion
- Record deletion.

Inquiry programs let the operator(s) enter a key field on a display, and then the programs display certain information from a corresponding record in a master file.

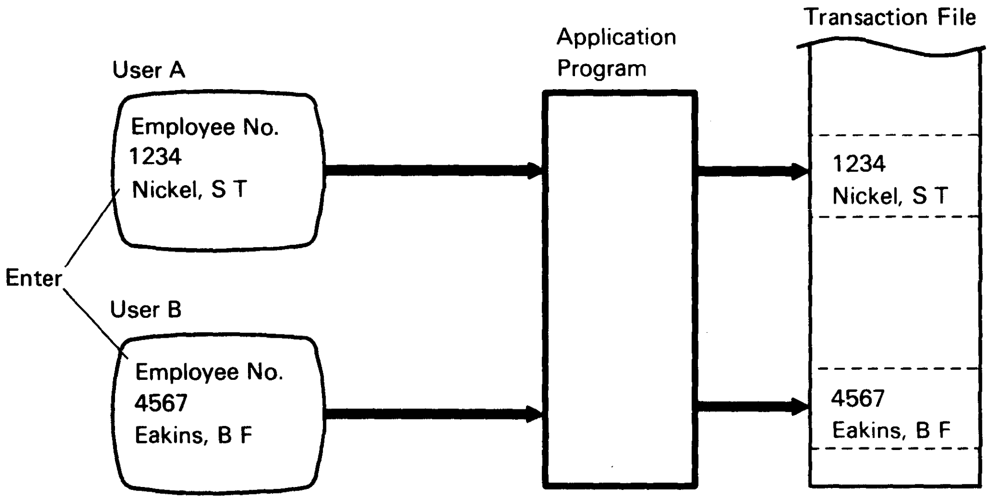
Inquiry



Maintenance programs let the operator(s) change or add records in an existing master file.

Simple data entry is like typing index cards, but with WSU the operator is prompted for the information, and the program takes care of putting it in the proper positions in the transaction file record.

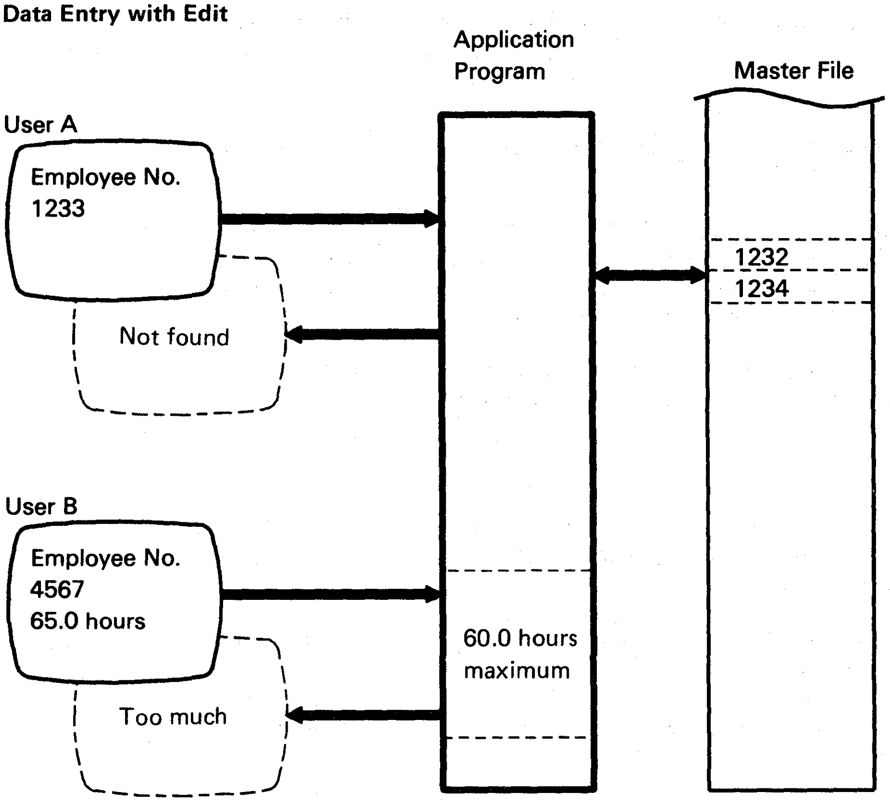
Simple Data Entry



Data entry with edit has the WSU program check to see if the typed data is correct. Editing can be done by:

- Checking the master file to verify entries (such as whether a master record exists for that transaction record, or whether overtime hours were entered for an employee on salary)
- Designing some standards into the program (such as whether overtime hours are excessive).

Data Entry with Edit



Data entry with update lets an operator also retrieve records from a transaction file and make changes to those records.

Record review lets an operator review transaction records entered, during the current session or during a previous session, from that display station.

Record insertion lets an operator place records *within* the chain of transaction records maintained for that display station.

Record deletion lets an operator remove records from the chain of transaction records maintained for that display station.

What WSU Is Not Designed to Do

WSU is intended only as a data entry/update, maintenance, and inquiry utility; it does not produce printed reports. So printed reports based on master or transaction files must still be produced using DFU or programs written in RPG or another programming language.

WSU Files

Files and arrays, including considerations for coding them, are described in Chapter 4, *Coding Files*, and Chapter 5, *Coding Arrays*.

Arrays

WSU allows you to search arrays, at the time your program is run, for a uniquely identified data item. WSU arrays are described on E-(array) specifications and created by C-(processing) specifications. Up to 75 execution-time arrays can be used in a WSU program.

Master Files

WSU uses the same master files, direct or indexed, as other languages and utilities.

Your system may have DDM (Distributed Data Management). DDM lets WSU programs use master files that are on a remote system. (A file on a remote system is referred to as a remote file.) For a brief description of DDM, see Appendix B.

A combined total of up to 20 remote and local master files can be referenced or updated in a WSU program.

Transaction Files

A WSU transaction file is a direct file, and only one is used per program. Also, only one program should work with any given transaction file.

ur system has DDM, your WSU program may use a remote transaction file.

The data in a transaction file can come from any or all of three sources:

- From operators, responding to prompts on the displays
- From related master files
- From the program.

The records in your transaction file can be of two types: detail and header.

- *Detail* records contain the information for one transaction.
- *Header* records contain information that applies to all the detail records associated with them.

Transaction files also contain, in addition to your data, some control information that WSU puts in for its own use. The control information includes:

- The *job control record*
- A *work session control record* for each display station

- A 13-position *trailer* field that WSU adds to the end of each data and control record, which contains pointers to form a logical chain of the records associated with each display station.

What WSU Requires to Generate an Application

WSU uses three things to generate an application program: a file definition, a source program, and a procedure command.

The File Definition

The *file definition* describes the records in the transaction and master files. The file definition for WSU consists of:

- A *F-(file description) specification* for each file, describing the physical characteristics and the organization of the file
- *I-(input) specifications* for each record type in the file, describing the format of the records. I-specifications also describe local data area fields and session-level fields.

These specifications must be stored in one or more source members that are different from the member that contains the WSU source program. There are several ways in which file definitions can be stored in a library. You can have:

- A separate source member for each file's F- and I-specifications
- One source member that contains *all* of the necessary F- and I-specifications
- Two or more source members with some members containing the F- and I-specifications for two or more files
- A source member that contains an entire RPG II source program. From that source member, WSU uses only the F- and I-specifications.

The T-specification in the WSU source program indicates which file definition source member contains the F- and I-specifications for the transaction file; the M-specifications in the WSU program indicate which file definition source members contains the F- and I-specifications for the master files as well as the I-specifications that describe local data area and session-level fields.

Chapter 12, *F-(File Description) and I-(Input) Specifications*, explains the entries on F- and I-specifications; Chapter 8, *Entering and Generating a WSU Program*, describes entering the F- and I-specifications using SEU.

The WSU Source Program

The *source program* consists of instructions to process data and to format the displays.

- The *J-(job) specification* describes some characteristics about the program you are writing.
- The *E-(array) specifications* describe any execution-time arrays that will be used for the job.
- The *T-(transaction file) specification* describes the transaction file (if any) that will be used for the job.
- The *M-(master file) specifications* describe any master files that will be used for the job.
- The *S-(display control) specifications* describe some overall characteristics of each display presented by the program.
- The *D-(field definition) specifications* describe the data that will be part of each display.
- The *C-(processing) specifications* describe any operations that are to be performed on the data as well as the direction the program should take in a given situation.

Chapter 4, *Coding Files*, discusses considerations for using files in your WSU program; Chapter 15, *T-(Transaction File) and M-(Master File) Specifications*, gives detailed information on completing each appropriate column of those specification sheets.

Chapter 5, *Coding Arrays*, discusses considerations for using arrays in your WSU program; Chapter 14, *E-(Array) Specifications*, gives detailed information on completing each appropriate column of that specification sheet.

Chapter 6, *Coding Displays*, discusses considerations for using displays in your WSU program; Chapter 16, *S-(Display Control) and D-(Field Definition) Specifications*, gives detailed information on completing each appropriate column of those specification sheets.

Chapter 7, *Coding Processing*, discusses considerations for using processing in your WSU program; Chapter 17, *C-(Processing) Specifications*, gives detailed information on completing each appropriate column of those specification sheets.

Chapter 8, *Entering and Generating a WSU Program*, tells you how to enter the specifications into the system.

WSU Procedure Command

When you have created and stored the file definition and the source program, you generate your program by entering the WSU procedure command. The command and its parameters are discussed in detail in Chapter 8, *Entering and Generating a WSU Program*.

What WSU Produces

When WSU builds, or generates, an application program from the file definition and source program you provide, it produces:

- A WSU *program* that is stored as a subroutine member (not as a load member)
- A *load member* containing the display formats
- A *procedure* that operators use to run the program/job
- A *generation printout* (similar to those produced by other programming languages) that lists the program instructions as well as any errors that WSU may have identified.

If WSU finds severe or **terminal** errors in your program, it gives you only the source printout identifying those errors and WSU ends. The output from a WSU program generation is described in Chapter 8, *Entering and Generating a WSU Program*, and in Chapter 9, *Testing and Debugging a WSU Program*.

Figure 1-1 shows the steps that WSU goes through to produce the procedure used to run WSU programs.

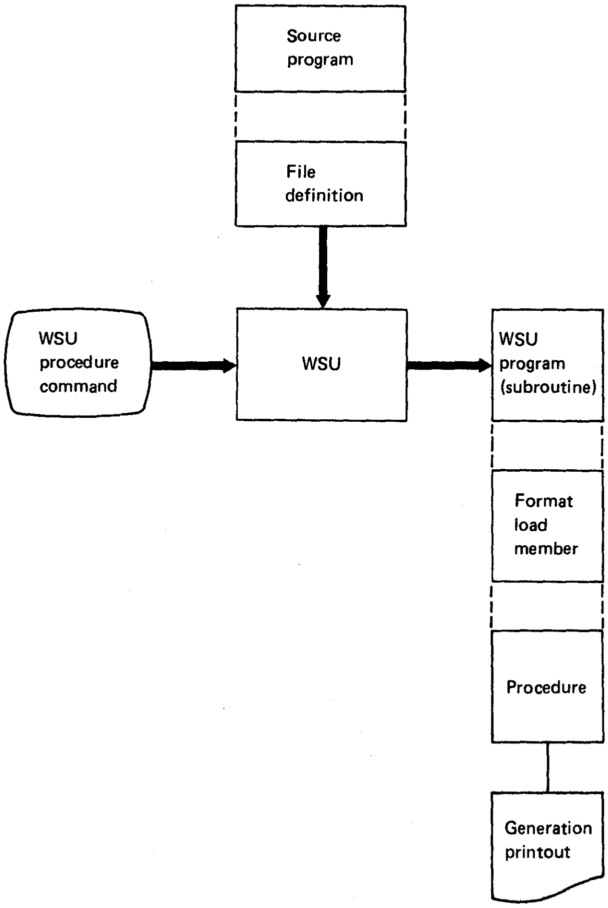


Figure 1-1. WSU Program Generation

Chapter 2. How WSU Works

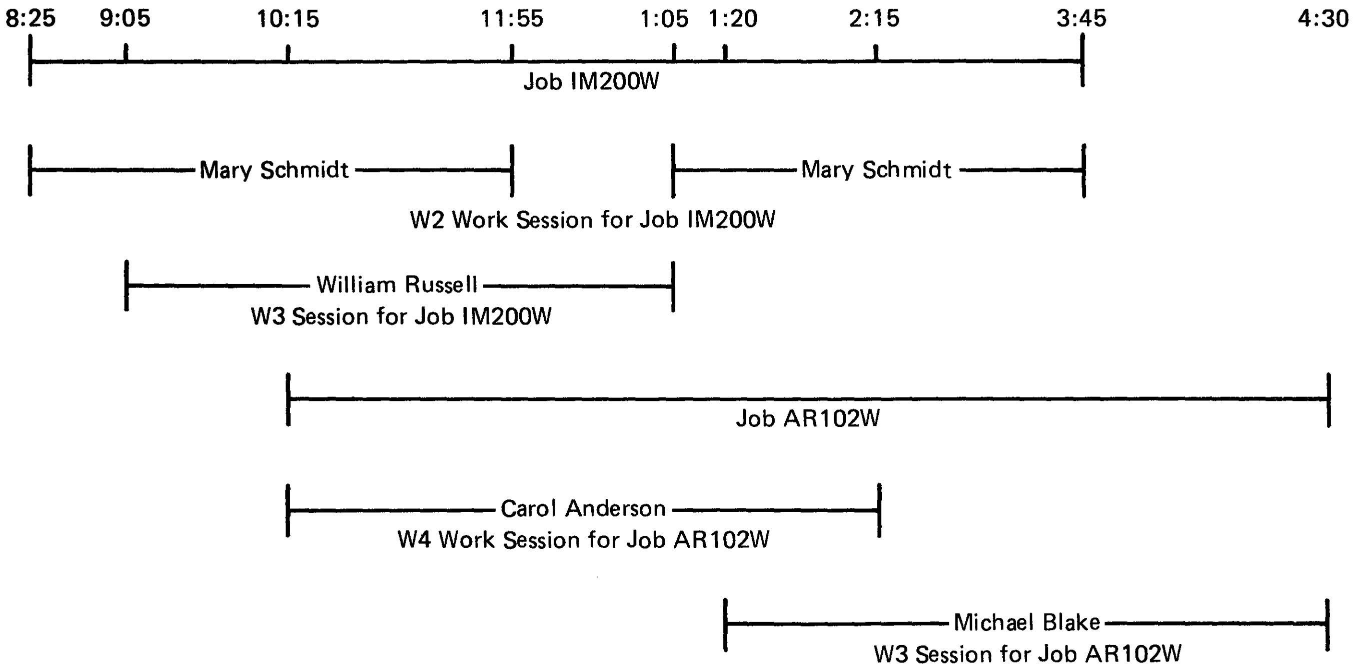
WSU Jobs and Work Sessions

A WSU *job* is the entire period starting when the first user (display station) signs on to a particular application program until the time the last user signs off. More than one WSU job (for example, an order entry program and an inventory inquiry program) can run at the same time.

A WSU *work session* is the period starting when a particular user (display station) signs on to a job until the time that particular user signs off. *Several* work sessions can begin and end during the course of a job, because WSU programs can be designed to allow more than one user at the same time.

Following is an example of WSU jobs and work sessions during a business day:

8:25 a.m.	Mary Schmidt signs on to a WSU program named IM200W from display station W2.	Begins job and session
9:05 a.m.	William Russell signs on to program IM200W from display station W3.	Begins session
10:15 a.m.	Carol Anderson signs on to a WSU program named AR102W from display station W4.	Begins job and session
11:55 a.m.	Mary Schmidt signs off program IM200W from station W2.	Ends session only (because W3 is still active)
1:05 p.m.	Mary Schmidt again signs on to program IM200W from station W2.	Begins another session
	William Russell signs off program IM200W from station W3.	Ends session only (because W2 is again active)
1:20 p.m.	Michael Blake signs on to program AR102W from station W3.	Begins session
2:15 p.m.	Carol Anderson signs off program AR102W from station W4.	Ends session
3:45 p.m.	Mary Schmidt signs off program IM200W from station W2.	Ends session and job
4:30 p.m.	Michael Blake signs off program AR102W from station W3.	Ends session and job


8:25
9:05
10:15
11:55
1:05
1:20
2:15
3:45
4:30
Job IM200W
Mary Schmidt
Mary Schmidt
W2 Work Session for Job IM200W
William Russell
W3 Session for Job IM200W
Job AR102W
Carol Anderson
W4 Work Session for Job AR102W
Michael Blake
W3 Session for Job AR102W

WSU Modes Of Operation

Modes are the four different ways in which WSU programs can work with a *transaction* file:

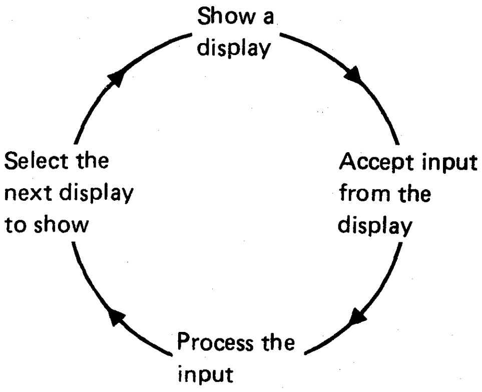
- If the operator is simply adding records to the transaction file by entering data in response to prompts (data entry), the program is said to be in *enter mode*. Enter is the initial mode and the standard mode when a program is run.
- If the operator is looking at or updating records previously entered in the transaction file (record review), the program is in *review mode*.
- If the operator is inserting records between records that already exist in the transaction file (record insertion), the program is in *insert mode*.
- If the operator is removing records from a transaction file (record delete), the program is in *delete mode*.

Operators change from one mode to another, if the program allows for such changes, by using command keys.

Enter Mode Program Cycle

The basic cycle for a WSU program (that is, for enter mode) is:

1. Show a display.
2. Accept input from the display.
3. Process the input.
4. Select the next display to **show**.
5. Return to 1.



WSU programming logically fits this cycle because you define a display (on S- and D-specifications), then define the processing that should occur for the display (on WSU C-specifications). And you define the displays and processing in your program in the order that you want the displays to appear.

Enter Mode Processing Levels

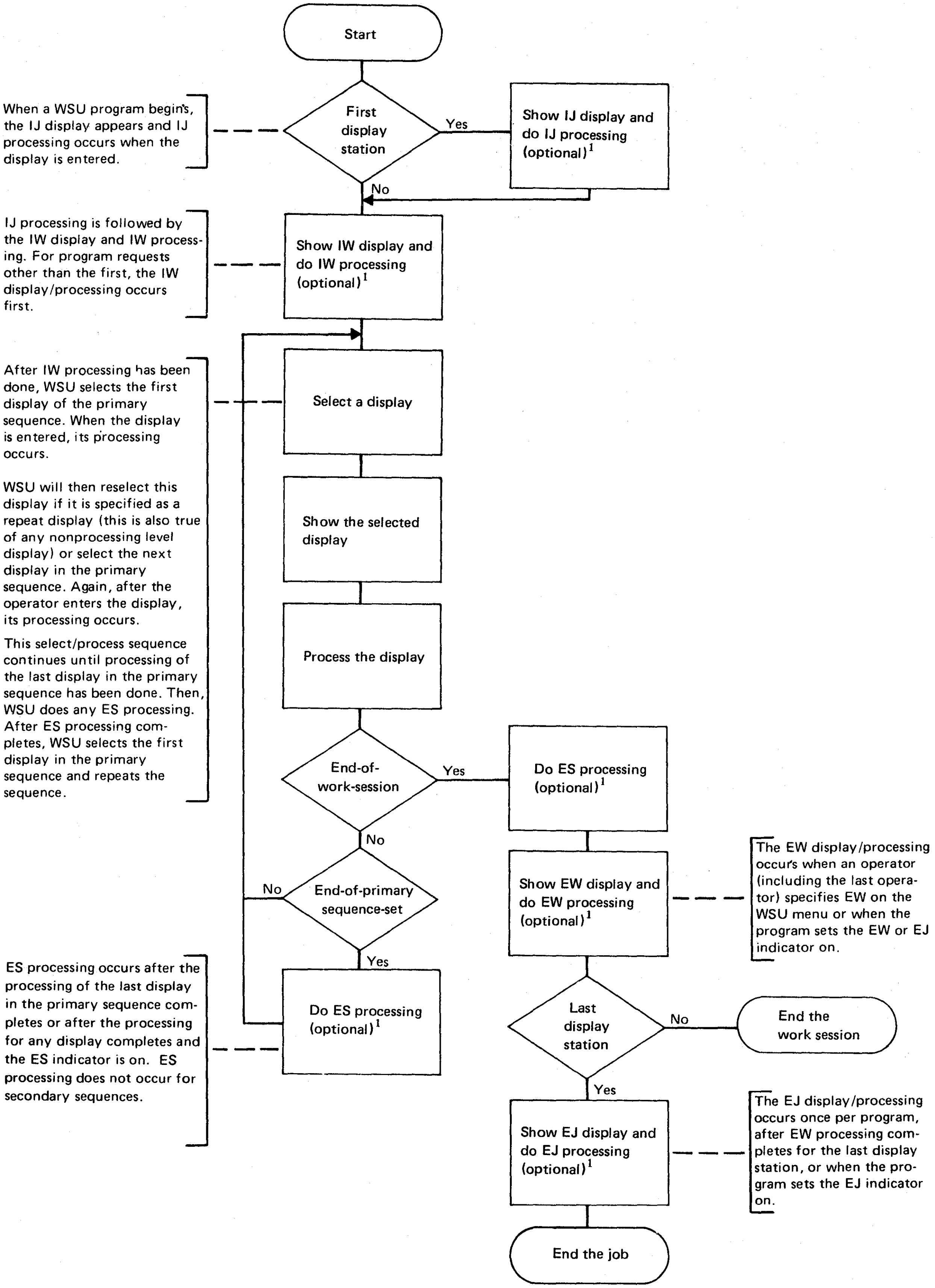
Processing levels are steps in the enter mode program cycle (see Figure 2-1) that occur automatically at specific times:

- *Job initiation (IJ)* occurs once, when the first display station calls the program.
- *Work session initiation (IW)* occurs once for each display station (including the first) before processing for that display station begins.
- *End of sequence set (ES)* occurs whenever a primary display sequence ends. (See *Enter Mode Display Sequences* later in this chapter.)
- *End of work session (EW)* occurs once for each display station (including the last) when the display station ends its use of the program.
- *End of job (EJ)* occurs once, when the last display station ends its use of the program.

You can ignore these steps in the cycle if you wish, or you can tie special processing and/or displays to them as follows:

- IJ, IW, EW, and EJ processing levels
  - One display and associated processing (S-, D-, and C-specifications) or
  - One display with no associated processing (S- and D-specifications only) or
  - Processing with no associated display (C-specifications only).
- ES processing level
  - Processing with no associated display (C-specifications only).

Refer to Chapter 6, *Coding Displays,* and Chapter 7, *Coding Processing,* for explanations of how to associate processing levels with displays and processing.



When a WSU program begins, the IJ display appears and IJ processing occurs when the display is entered.

IJ processing is followed by the IW display and IW processing. For program requests other than the first, the IW display/processing occurs first.

After IW processing has been done, WSU selects the first display of the primary sequence. When the display is entered, its processing occurs.

WSU will then reselect this display if it is specified as a repeat display (this is also true of any nonprocessing level display) or select the next display in the primary sequence. Again, after the operator enters the display, its processing occurs.

This select/process sequence continues until processing of the last display in the primary sequence has been done. Then, WSU does any ES processing. After ES processing completes, WSU selects the first display in the primary sequence and repeats the sequence.

ES processing occurs after the processing of the last display in the primary sequence completes or after the processing for any display completes and the ES indicator is on. ES processing does not occur for secondary sequences.

The EW display/processing occurs when an operator (including the last operator) specifies EW on the WSU menu or when the program sets the EW or EJ indicator on.

The EJ display/processing occurs once per program, after EW processing completes for the last display station, or when the program sets the EJ indicator on.

[1] *Optional* means the program need not include a display and/or processing instructions for this level.

Figure 2-1. WSU Enter Mode Processing Levels

Enter Mode Display Sequences

You can define as many as 245 displays in a WSU program. In enter mode, displays can be sequenced (grouped in a particular order) so that when one display completes processing, the next one in sequence is automatically presented and when the last display of the sequence completes processing, the first display is automatically presented again.

Once a sequence begins, a work session stays in that sequence until the program or the operator selects a display that is out of the sequence. The program can modify display sequences by using PUTS, MSG, and IMSG operations (see Chapter 7, *Coding Processing*, and Chapter 18, *Operation Codes)*.

The operator can modify display sequences either by pressing the Bypass Display command key (Cmd 2) or by pressing the WSU Display command key (Cmd 1) and then responding to the WSU display. (Chapter 10, *Running a WSU Program*, describes the keys and the WSU display).

WSU recognizes three types of display sequences: primary, secondary, and nonsequenced. (IJ, IW, EW, and EJ displays are not included).

Primary Display Sequence

The primary display sequence is placed first in the source program (unless you are using IJ and/or IW displays). WSU *requires* one (and *allows* only one) primary sequence per program; however, this sequence can be as brief as one display.

Secondary Display Sequences

Secondary display sequences are optional; if used, they immediately follow the primary sequence in the source program. You can include more than one secondary sequence per program.

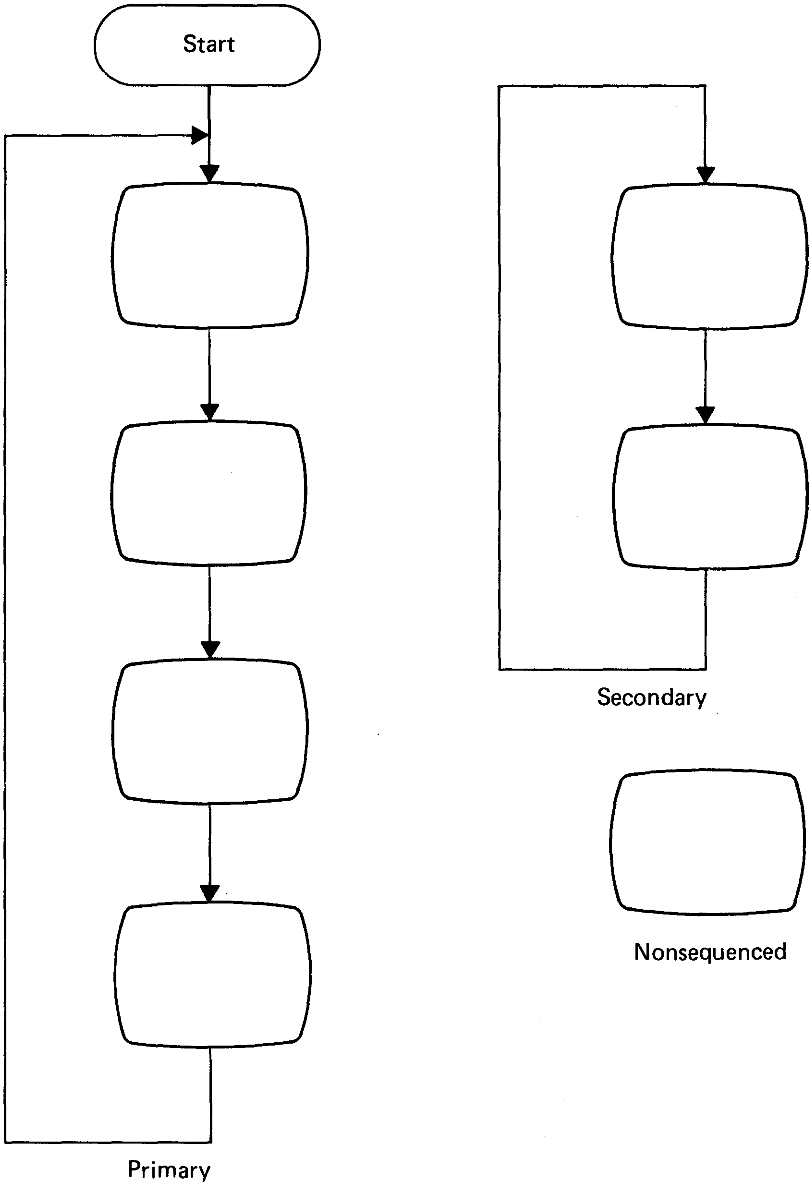
Nonsequenced Displays

Nonsequenced displays are not included in either the primary sequence or any of the secondary sequences. They must follow all sequenced displays in the source program. You can include more than one nonsequenced display per program, and you can use a primary sequence and nonsequenced displays with no intervening secondary sequences.

Defining Display Sequences

Display sequences are defined by the *Start Sequence* and *End Sequence* entries of the S-specification and by their place in the source program. See Chapter 6, *Coding Displays,* and Chapter 16, *S-(Display Control)* and *D-(Field Definition) Specifications* for further information.

Types of Display Sequences



Review Mode

When you code your WSU program to allow review mode for a transaction file, operators can look at, or look at and change, those records in the transaction file chain for the current work session. That is, if an operator were at display station W1, he could review any records in the transaction file that had been entered from W1, whether he had just entered them or whether he or someone else had entered them earlier in the day or week. (It is possible to allow operators to also review records entered from display stations other than the one they are using; see *Universal Work Session Selection Authorization* in Chapter 8.)

You code your WSU program to allow review mode by designating one or more displays in your program as *review-capable*; that is, a display that can be used to review records. You do this by specifying the record types that can be reviewed using that display as the *Review Mode Record Identifying Indicators* entry (columns 48 through 53) of the appropriate S-specification. (For more information, see Chapter 6, *Coding Displays*, and Chapter 16, *S-(Display Control)* and *D-(Field Definition) Specifications*.

You can also code certain conditions and operations for review mode on the display's C-specification. (For more information, see Chapter 7, *Coding Processing*, and Chapter 17, *C-(Processing) Specifications*.

Review Mode Variations

Review Only

If you want operators to be able to look at records but to make no changes to them, there are several methods you can use to code processing for the review-capable displays:

- You can code a GOTO operation that branches to the end of the processing when the RV (review mode) indicator is on.
- You can condition PUT operations to occur only when the RV indicator is *not* on.
- You can define a *preprocessed* display where, when the RV indicator is on, a SETON operation turns on another indicator (01 through 89) to protect all input fields; then a PUTS operation presents the review display. You also need to specify the indicator for the *Protect Field* entry on the D-specification. Preprocessing is described in Chapter 7.

Review with Update

If you want operators to be able to change records as well as look at them, you can simply use the RV indicator to condition the operations on the C-specifications that should (or should not) occur during review.

Review with Master File Update

If you want to allow changes to the master file record at the same time that the transaction file is being reviewed/changed, the review display should be a preprocessed display. That way, in the preprocessing, certain operations can be done on the master file record before the transaction file record is displayed for review and the operators have a chance to change the data. See *Updating Master File Records During Review Mode* later in this chapter for some things to consider when coding reviews of this kind.

Review Mode Displays

WSU does *not* automatically generate separate displays for review mode from existing enter-mode display formats. You must either code and condition your enter-mode displays to be used for review as well *or* code separate displays for review mode that are in a separate sequence of their own.

Separate Review Mode Displays

Although coding additional displays may at first seem like a lot of extra effort, you will probably find that your coding is actually simpler when you keep the enter displays and processing and review displays and processing separate.

Combined Enter/Review Mode Displays

You may still prefer to use the same display for both entry and review. If so, for each input field, the *Output Data* entry (columns 23 and 24) on the D-specification must be conditioned by an indicator that is turned on when review mode starts.

WSU does this for you automatically if you have nothing else coded for that entry; it inserts indicator 99 (one of its reserved indicators) as the value for columns 23 and 24 and then turns indicator 99 on when review mode starts.

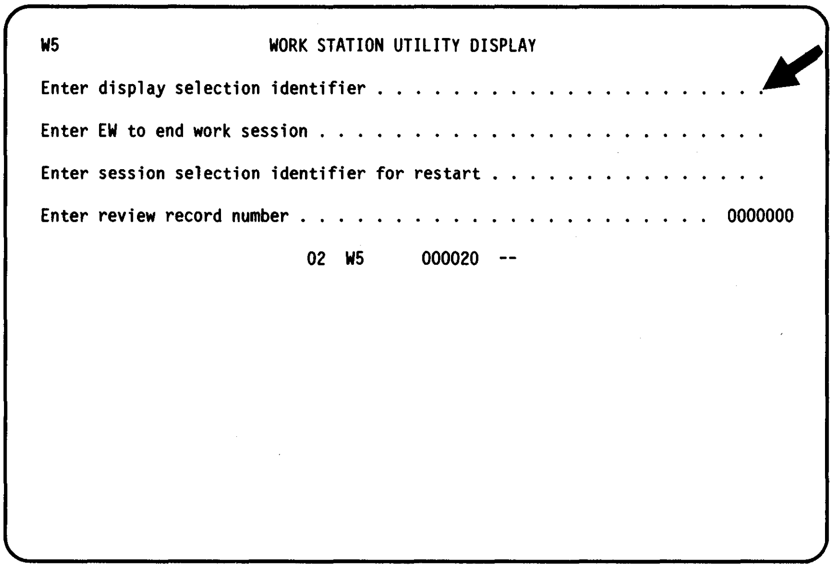
However, if you have already coded an indicator for the *Output Data* entry, WSU cannot use its indicator. In that case, you will have to use a SETON operation in the C-specifications to turn your indicator on when review mode starts (indicator RV is on) if you want that field displayed for review.

Display Sequences in Review Mode

Enter mode's automatic display sequence does not apply to review mode, nor do the required or repeated displays. Instead, the program and/or the operator control display selection.

The operator can control display selection by either:

- Using the Bypass Display command key (Cmd 2) to get the next review-capable display
- If he knows the format ID of the display he wants to select, using the WSU Display command key (Cmd 1) to get the menu and then entering the format ID on the first line of the menu.



```
W5                     WORK STATION UTILITY DISPLAY

Enter display selection identifier . . . . . . . . . . . . . . . . . . . . .

Enter EW to end work session . . . . . . . . . . . . . . . . . . . . . . . .

Enter session selection identifier for restart . . . . . . . . . . . . . . .

Enter review record number . . . . . . . . . . . . . . . . . . . . . . 0000000

                             02  W5      000020  --
```

When an operator finishes reviewing a record, he can either press the Enter key to process that display or press a command or function key to request another WSU function.

If he presses the Enter key, all the input fields are given to the program, and processing for the display begins. When processing is finished, WSU sends a message, and the operator can do one of the following:

- Select another record or display for review
- Begin insert or delete mode
- Resume enter mode
- Select a new work session (if authorized to do so)
- End the work session.

If he presses an enabled WSU command or function key instead of the Enter key, the processing for that display does not happen (although preprocessing, if any, does).

Review Mode Functions

An operator starts review mode when, during enter mode, he presses the appropriate command or function key to request one of the following review functions:

- Page backward
- Page forward
- Review by relative record number
- Resume review.

Page Backward

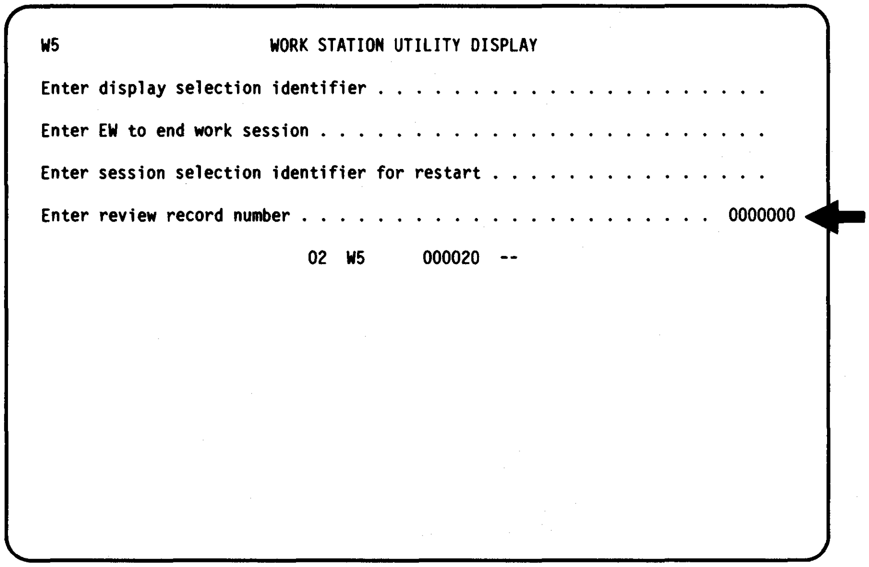
An operator can review the previous record in the chain by pressing the Page Backward Record function key (Roll $\underline{R}$). Or, if the file contains header records as well as detail records, he can review the previous record by pressing the Page Backward Group command key (Cmd 5). (Header records are designated by an entry in the *Header Record Identifying Indicator field* of the T-specification; for more information, see Chapter 4, *Coding Files*, and Chapter 15, *T-(Transaction File)* and *M-(Master File) Specifications*.) If the operator tries to page backward past the first record of the chain, WSU displays an error message. However, the operator can *wrap around* the file to the *last* record (or header record) in the chain by pressing the Error Reset key in response to the message, then pressing the appropriate Page Backward key again.

Page Forward

An operator can review the next record in the chain by pressing the Page Forward Record function key (Roll ↑). Or, if the file contains header records as well as detail records, he can review the next *header* record by pressing the Page Forward Group command key (Cmd 6). If the operator tries to page forward past the last record of the chain, WSU sends an error message. However, the operator can *wrap around* the file to the *first* record (or header record) in the chain by pressing the Error Reset key in response to the message, then pressing the appropriate Page Forward key again.

Review by Relative Record Number

If the operator knows the relative record number of the record he wants to review, he can press the WSU Display command key (Cmd 1) and specify that number on the fourth line of the WSU display.



```
W5                    WORK STATION UTILITY DISPLAY

Enter display selection identifier . . . . . . . . . . . . . . . . . . . . .

Enter EW to end work session . . . . . . . . . . . . . . . . . . . . . . . .

Enter session selection identifier for restart . . . . . . . . . . . . . . .

Enter review record number . . . . . . . . . . . . . . . . . . . . . . 0000000

                         02  W5      000020  --
```

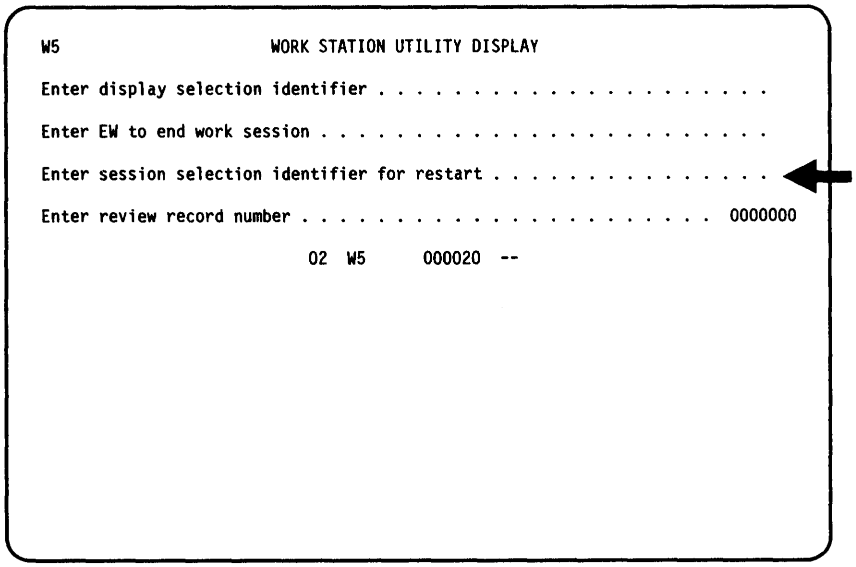
WSU places the specified number in a reserved field named *RLRN (Chapter 3 describes the reserved fields and their uses).

This function is useful if you want to *batch-edit* the transaction file; that is, after the data is entered, you use a separate edit program (or manual process) to verify all the data, then the operators use the data entry program again to make any needed corrections to specific records.

You can use the WSU Extract procedure (WSUTXEX) to select and print, in logical order, the chain of records for a work session, including the relative record numbers. The WSU Extract procedure is described in Chapter 4, *Coding Files*, and in Chapter 19, *File Processing Procedures*.

If the operator enters the relative record number of a record that is not in his work session chain, WSU displays an error message (but does *not* turn on the record-not-found indicator) and places the identifier of the *owning* session in a reserved field named *USID.

WSU also shows the ID of the record's session chain as the default selection on the third line of the WSU display.



```
W5                     WORK STATION UTILITY DISPLAY

Enter display selection identifier . . . . . . . . . . . . . . . . . . . . .

Enter EW to end work session . . . . . . . . . . . . . . . . . . . . . . . .

Enter session selection identifier for restart . . . . . . . . . . . . . . .

Enter review record number . . . . . . . . . . . . . . . . . . . . . . 0000000

                          02  W5      000020  --
```

You can modify the generated procedure for your program (see Chapter 8) to allow operators to select work sessions other than their own. If you do so, the operator can, after getting the message that the requested record is not his, press the WSU Display command key (Cmd 1) again and select the session that the record belongs to.

Resume Review

If the operator wants to see the record he most recently reviewed again, he can press the Resume Review Mode command key (Cmd 15) from the WSU display or from any display in any mode. WSU knows which record was most recently reviewed because, when a record is reviewed, WSU places its relative record number in a reserved field named *RLRR (Chapter 3 describes the reserved fields and their uses).

WSU's Role in Review Mode

When an operator requests a review function by pressing one of the appropriate keys (see *Review Functions* earlier in this section), WSU does the following things:

- Sends a message telling the operator that he is now in review mode
- Saves all the mode-level fields and indicators that were being used in enter mode (so they can be restored later when the operator resumes enter mode)
- Sets the mode-level fields to blanks or zeros and the mode-level indicators off (so they can be used for review and processing)
- Sets on the review mode indicator (RV) and indicator 99 (reserved indicator)
- Retrieves the appropriate record and determines its type
- Searches the S-specifications in the program (from first to last) for the first display that allows review of that record type
- Shows the operator the requested record using the appropriate review-capable display
- Places the relative record number of the record being reviewed in a reserved field named *RLRN, thus designating the record as the *current record* and using it as a reference point for the next paging request or chain-dependent operation (like GETNR)
- Places the relative record number of the record previously reviewed in a reserved field named *RLRR, thus designating the record as *most recently reviewed* and using it as a reference point for insert, delete, or resume review functions
- Issues a message if one of the following situations occur:
    - The record could not be identified
    - No review-capable display could be found for the requested record's type
    - An attempt was made to page past the end or the beginning of the chain
    - The requested relative record number is not for a record in the requester's chain.

Updating Master File Records During Review Mode

Master file records can be read and rewritten while your program is in review mode; however, you should not code a PUTS, MSG, or IMSG operation, nor a GET operation for a different file, between the point the master file record is read and the point it is subsequently rewritten. Also, a master file record must be read and rewritten in the same mode, without an intervening mode change or an intervening review, delete, or insert request. That is, a program cannot read a master file record in enter mode and write that record in review mode.

For a review that allows master file update, the review display should be a preprocessed display. During preprocessing, certain operations can be done to the master file record before the reviewed transaction file record is displayed and operators have a chance to change the data.

Remember that an operator is not required to *enter* a reviewed record, so preprocessing for the review display could happen but processing not happen. Therefore, information from the master file record should be saved in the preprocessing and changed in the processing.

Once the master file record data is read and saved, you should code a PUTS operation to display the reviewed record. Then, when the operator updates the data and enters the display, processing resumes with the operation that follows the PUTS operation.

You should condition operations with the RV indicator to read the proper master file record and to use the reviewed transaction file record data to restore the master file fields to their original status before the update was made. (This restoring is done by reversing the updates, for example, subtracting data that was added.) The processing that follows the master file update should do the same processing that was done for the initial entry of the record.

Insert Mode

When you code your program to allow insert mode for a transaction file, operators can logically insert all types of records (except header records) into the transaction file chain for the current session. That is if an operator is at display station W2, he can insert records into the chain of records that have been entered at W2. (It is possible to allow operators to also insert records into chains of other work sessions; see *Universal Work Session Selection Authorization* in Chapter 8.)

Master file records can be read and written while the program is in insert mode; however, they must be both read and written in the same mode.

If you want to allow insert mode in your program, you must also allow review mode (see the previous section for information on coding and using review mode). To allow insert mode in your program, you must designate one or more displays as *insert-capable* that is, a display that can be used to insert records (normally these would be the enter-mode displays). You do this by specifying the record types that can be inserted using that display as the value for the *Insert Mode Record Identifying Indicators* entry of the S-specification. (For more information, see Chapter 6, *Coding Displays*, and Chapter 16, *S-(Display Control)* and *D-(Field Definition) Specifications*.

An operator starts insert mode when, during review mode, he displays the record that will precede the inserted records and then presses the Insert Mode command key (Cmd 4).

Display Sequence in Insert Mode

Enter mode's automatic display sequence does not apply to insert mode, nor do the required or repeated displays. Instead, the program and/or the operator control display selection.

The operator can use the Bypass Display command key (Cmd 2) to get the next insert-capable display for a record. When the operator finishes typing the information to be inserted, he presses the Enter key to give the input fields to the program and begin processing. When processing completes, WSU sends a message and the operator can then do one of the following:

- Insert another record
- Resume enter mode
- Resume review mode
- Select a new work session (if authorized to do so)
- End the work session.

Inserting a Group of Records

To allow a *group* of records to be inserted, you can code the processing for an insert-capable display to handle multiple inserts using one of the following methods:

- For a display that is *not* preprocessed:
  - Code a PUTS operation, conditioned by the IN indicator, as the last operation in the C-specifications for that display.
- For a display that is preprocessed:
  - Code a GOTO operation, directed to the first operation in the preprocessing, and conditioned by the IN indicator, as the last operation in the C-specifications for that display.

Or, if you do not include such coding in your program, the operator can switch back and forth between review mode and insert mode for each record to be inserted, repeating the following steps:

1. Insert a record.
2. Press the Page Forward Record function key (Roll ↑) to display the inserted record.
3. Press the Insert Mode command key (Cmd 4).
4. Repeat the process.

WSU's Role in Insert Mode

Before starting insert mode, an operator must first be in review mode and must review the record that will precede the inserted record. Then, when an operator starts insert mode by pressing the Insert Mode command key (Cmd 4), WSU:

- Reads the displayed record and determines its type
- Searches the S-specifications in the program (from first to last) for the first display that allows insertion after that record type
- Presents that display to allow an operator to insert a record
- Sets on the insert mode indicator (IN) so that input fields also become output fields.

Except for indicator IN and reserved field *RLNO, the mode-level fields and indicators remain as they were in review mode.

When an operator enters an inserted record, WSU:

- Places the inserted record after the last nonblank record in the file
- Sends a message to the operator.

When an operator ends insert mode, WSU:

- Updates the trailers to *link* the inserted record into the logical chain
- Sets indicator IN off.

Delete Mode

When you code your program to allow delete mode for a transaction file, operators can logically delete any types of records from the transaction file chain for the current session. That is, if an operator is at display station W2, he can delete records from the chain of records that have been entered at W2. (It is possible to allow operators to also delete records from chains of other work sessions; see *Universal Work Session Selection Authorization* in Chapter 8).

If a header record is deleted, all the records associated with the header record remain linked together, and the entire group of associated records is logically removed from the transaction file chain.

If you want to allow delete mode in your program, you must also allow review mode, because WSU deletes the *most recently reviewed* record (that is, the one whose relative record number is in the reserved field named *RLRR). See the previous section for information on coding/using review mode.

An operator starts delete mode when, during review mode, he presses the Delete Mode command key (Cmd 14). WSU displays the most recently reviewed record for verification and a message that delete mode is active. The operator can then (after responding to the message) either:

- Cancel the delete request by doing one of the following:
  - pressing the Accept-Sequence-Error command key (Cmd 13)
  - requesting a different mode
- Confirm the delete request by pressing the Enter key.

WSU's Role in Delete Mode

When an operator starts delete mode, WSU:

- Sends a message telling the operator that he is now in delete mode
- Determines which is the most recently reviewed record and determines its type
- Searches the S-specifications in the program (from first to last) for the first display that allows review of that record type
- Shows the most recently reviewed record using that display to allow the operator to verify the delete request
- Sets on the delete mode indicator (DL).

If delete mode was started from enter mode, WSU saves all the mode-level fields and indicators and sets the mode-level fields to blanks or zeros and the mode-level indicators off.

If delete mode was started from review mode, the mode-level fields and indicators (except for indicator RV) remain as they were in review mode.

When an operator deletes a record, WSU:

- Updates the trailers to remove the deleted record from the logical chain
- Sets indicator DL off
- Sends a message to the operator.

Display Station Fields and Job-level Fields

WSU uses an area in main storage in which it keeps all of the fields defined in your program. It classifies these fields as:

- Display station fields
- Job-level fields.

Display Station Fields

Because a WSU program is an MRT (multiple requester terminal) program, it can accept input from several display stations, one station at a time. The *current display station* is the one from which the program is accepting input.

Each display station has a set of fields that are initialized when a work session is started at the display station. These fields are moved into the field area in main storage when their display station becomes current and moved out of the field area when that display station is no longer current. These fields can be mode-level or session-level, but not mixed.

Session-Level Fields

Session-level display station fields retain their values when the operating mode changes.

You can define session-level fields on the I-specifications and on the M-specifications. In addition, some of WSU's reserved fields are session-level fields. (See Chapter 3 for information on reserved fields and their uses.)

Mode-Level Fields

Mode-level display station fields are saved when the operating mode changes and then cleared to blanks or zeros for use during the new mode.

Mode-level fields are not explicitly defined on a specification; rather, all fields that are neither session-level nor job-level become mode-level fields by default. In addition, some of WSU's reserved fields are mode-level fields.

Job-Level Fields

Each WSU program has one set of job-level fields. These fields do not move to and from the field area in main storage; instead, they remain in the field area and are available to any active display station.

You can define job-level fields on the C-specification; their names must begin with an &. In addition, WSU's reserved date fields are job-level fields.

Display Station Local Data Area

Each display station has a *local data area* that you can use in your program by defining it on the I-specifications and M-specifications. WSU then automatically reads a display station's local data area when the station signs on and updates the area when the station signs off. The *Concepts and Programmer's Guide* describes the local data area and its uses.

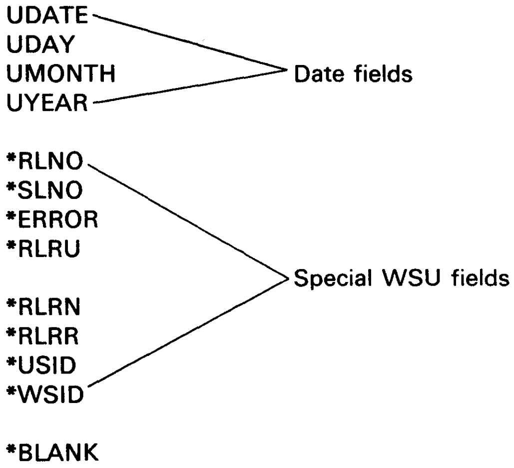
Chapter 3. Reserved Fields and Indicators

This chapter provides information about:

- Reserved Fields: Special fields provided by WSU
- Indicators: Internal switches used to tell when certain operations are to be performed.

Using Reserved Fields

Reserved fields are special fields that are supplied by WSU for specific functions. These reserved fields are:



UDATE, UDAY, UMONTH, and UYEAR are job-level fields (fields that remain in the field area in main storage); *RLNO, *SLNO, *ERROR, and *RLRU are mode-level fields; *RLRN, *RLRR, *USID, and *WSID are session-level fields (fields that retain their value when the operating mode changes); and *BLANK is a special field that is used only with the MOVE operation.

Do not begin *RLRN, *RLNO, *RLRU, *RLRR, *SLNO, *BLANK, *ERROR, *USID, or *WSID in column 7 of a continued line in a table of values since an asterisk (*) in column 7 indicates a comment line.

Date Fields

When the WSU program is initialized, WSU records the following information in the date fields:

- UDAY: Day
- UMONTH: Month
- UYEAR: Year
- UDATE: Program date if a program date has been specified; system date if a program date has not been specified.

You can use date fields:

- As output fields on displays
- In factor 1 or factor 2 for operations that allow numeric fields for these entries
- In a table of values for a COMP operation
- As GET fields on an M-specification.

*RLNO

*RLNO is a 6-byte numeric field that contains a relative record number and is a display station field. At job initiation prior to any transaction file add operation, *RLNO can be used to calculate the number of records that can be added to the file. After job initiation or after the first add operation during job initiation, *RLNO can be used to record the relative record number of the next record to be added to the transaction file for each work session.

At job initiation, *RLNO contains the largest reserved relative record number in the transaction file. Also, at work session initiation or whenever a record is added to the transaction file, *RLNO contains the relative record number of the *next* record to be added to the transaction file for the corresponding display station. This value will be different for each display station, depending on whether the display station is in enter mode or insert mode. If there is no transaction file, *RLNO contains zero during the program's running. In review mode, *RLNO is set to zero.

You can code *RLNO in factor 1 or factor 2 of operations that allow numeric fields for these entries. *RLNO can also be in a table of values for a COMP operation or a GET field on an M-specification. *RLNO cannot be an input field on a D-specification, or a result field.

*SLNO

*SLNO is the reserved field that contains the variable starting line number of a display. The S-specifications for such displays have a V entered for the starting line number.

*SLNO is a 2-byte numeric field with no decimal positions and is a display station field. This field has an initial value of 1 and assumes other values only as a direct result of C-specification processing. The contents of *SLNO are restored to their enter-mode value when the mode changes from review to enter. Other than this restoring, WSU does not change the contents of *SLNO.

During WSU running, the work session ends abnormally if the sum of *SLNO and the maximum line number for the display is greater than 24. The maximum line number is the larger of the number of lines to clear and the largest line number on the D-specification.

*SLNO can be coded wherever a field name is allowed, except on I-specifications and D-specifications. Also, *SLNO can be used in any processing function.

*ERROR

The *ERROR reserved field is a 4-byte numeric field that contains a 4-digit indication of errors that occur for transaction file operations (GETNR, GETNH, GETPR, GETPH, and PUT) and master file operations (GET, PUT, and PUTN). The *ERROR reserved field can be used to plan error recovery based on a specific error code.

*ERROR is set to zero when a file operation occurs. If an error indicator has been coded in columns 61 and 62 of the T-specification or M-specification and an error occurs that causes this indicator to turn on, WSU places one of the following error codes in *ERROR:

Code	**Meaning**
0004	An I/O error occurred for a GET operation.
0005	A record-not-found error occurred but a not-found indicator was not coded.
0006	The record key contains hex FF for a GET operation.
0007	An I/O error occurred on a GET operation before a PUT operation.
0008	A PUT operation for a file update occurred, but a successful GET operation did not come before it.
0009	An I/O error occurred on a PUT operation.
0012	An attempt was made to insert a transaction file header record with a PUT operation.
0029	The transaction file was full for a GET operation that occurred before a PUT operation.
0030	The transaction file record number exceeds 65,535.
0033	The key of the record being updated by means of a PUT operation differs from the key of the record read by means of the previous GET operation.
0041	The end of file has been reached on a PUTN operation.
0042	The PUTN operation would duplicate an existing record key.
0043	The PUT operation would duplicate an existing alternate record key.
0044	An I/O error occurred for a PUTN operation.
0056	The PUTN operation would duplicate an existing alternate record key.
0057	The PUT operation would duplicate an existing record key.

You can code *ERROR in factor 1 or factor 2 of operations that allow numeric fields for these entries. *ERROR can also be in a table of field names for a COMP operation.

*RLRU

*RLRU is a 6-byte numeric field that contains the relative record number for the most recent successful input operation performed on the transaction file for a C-specification operation (GETNR, GETNH, GETPR, GETPH). The initial value of the field is zero.

You can code *RLRU in factor 1 or factor 2 of operations that allow numeric fields for these entries, or as a GET field on the M-specification.

*RLRN

*RLRN is a 6-byte numeric field that contains either the relative record number that an operator specified on the WSU display or the most recent value that the program placed in it. The contents of *RLRN are not set to zero when review mode begins. Also, the contents of *RLRN are not changed as records are reviewed by means of the Page Backward Group command key (Cmd 5) or the Page Forward Group command key (Cmd 6), or Page Backward Record (Roll ↓) or Page Forward Record (Roll ↑) function key. The contents of *RLRN are not destroyed at the end of a work session; they are passed to the next work session.

You can code *RLRN in factor 1, factor 2, or the result field of operations that allow numeric fields for these entries. You can also code it as a GET field on an M-specification.

*RLRR

*RLRR is a 6-byte numeric field that contains the relative record number for the most recent successful input operation performed on the transaction file when a review function is requested by the operator. The initial value of the field is zero.

You can code *RLRR in factor 1 or factor 2 of operations that allow numeric fields for these entries, or as a GET field on the M-specification.

*USID

*USID is a 2-byte alphameric field that contains either:

- Blanks for a new requester
- The session ID entered by the operator on the WSU display
- The session ID from the trailer of a WSU file data record or header record (after an I/O operation has been performed on the transaction file for a review request).

The first byte must be alphabetic; the second byte must be alphameric.

The contents of *USID are not destroyed at the end of a work session; they are passed to the next work session.

You can code *USID in factor 1, factor 2, or the result field of operations that allow alphameric fields for these entries.

*WSID

*WSID is the reserved field that contains the symbolic display station identifier.

*WSID is a 2-character alphameric field that is set to the symbolic display station ID when the display station first signs on.

You can code *WSID as factor 1 or factor 2 of a C-specification. *WSID cannot be used on D-specifications or I-specifications.

*BLANK

*BLANK can be used in factor 2 of the MOVE operation to set an alphameric result field to blanks.

INDICATORS

An indicator is an internal switch designated by two characters (two letters, two digits, or a combination of a letter and a digit). Indicators can be used to control whether certain operations are performed.

WSU automatically turns some of the indicators on and off to indicate the following conditions:

- The current or a waiting processing level
  - IJ: Job initiation
  - IW: Work session initiation
  - ES: End of sequence set
  - EW: End of work session
  - EJ: End of job
- The operating mode
  - RV: Review mode
  - IN: Insert mode
  - DL: Delete mode
- The command key an operator pressed
  - KG: Command key 7
  - KH: Command key 8
  - KI: Command key 9
  - KJ: Command key 10
  - KK: Command key 11
  - KL: Command key 12
  - KQ: Command key 16
  - KR: Command key 17
  - KS: Command key 18
  - KT: Command key 19
  - KU: Command key 20
  - KV: Command key 21
  - KW: Command key 22
  - KX: Command key 23
  - KY: Command key 24
- A reviewed record from the same current group as the record being entered (CG)
- A session restarted after an abnormal ending (RC) or after a record was logically deleted from the session chain being restarted (the transaction file might contain records which have been removed from the chain for the session)
- A session restarted after normal ending (RS).

You can turn some of the WSU indicators on and off in your WSU program. These indicators are:

AC	(accept command key) Off only.
AE	(accept sequence error) You can set this indicator on to allow operators to use the Accept-Sequence-Error command key to bypass required displays.
EJ	(end of job)
ES	(end of sequence set)
EW	(end of work session)
IS	(initiate transaction sequence) Off only.
JA-JN, JP-JY	(job)
KG-KL, KQ-KY	(command key)
PG	(program mode)
RC	(recovery of work session)
RP	(repeated display)
RS	(resume work session)
SA-SN, SP-SY	(session level)
U1-U8	(external) The operator can set these indicators on or off by means of the SWITCH OCL statement when the program is run or WSU sets them on or off when they are used as resulting indicators.
01-89	(display station)

Notice that the EJ, ES, EW, KG-KL, RC, RS, and KQ-KY indicators are in both categories, both WSU controlled and programmer controlled.

Job, Session, and Mode Indicators

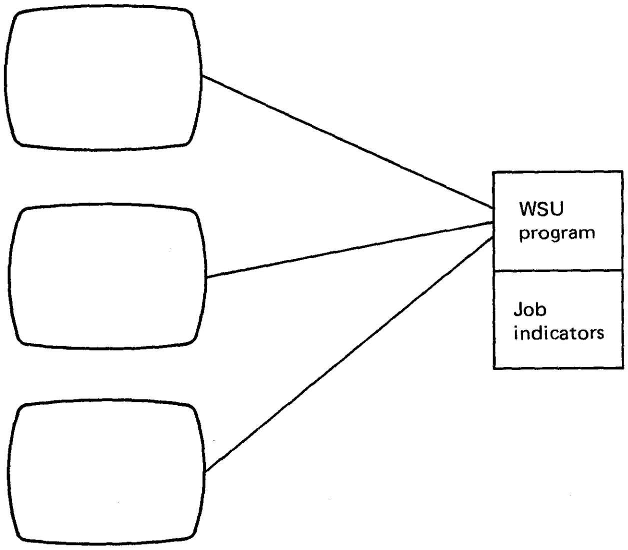
WSU maintains indicators for the job, for a session, and for an operating mode.

Job Indicators

The job indicators are:

EJ	(end-of-job processing level) Indicates the end of a WSU program.
IJ	(job-initiation processing level) Indicates the beginning of a WSU program.
JA-JN, JP-JY	(job) Provide session-to-session saving of indicator settings.

As the following shows, WSU maintains one copy of these indicators for all display stations using the same program.



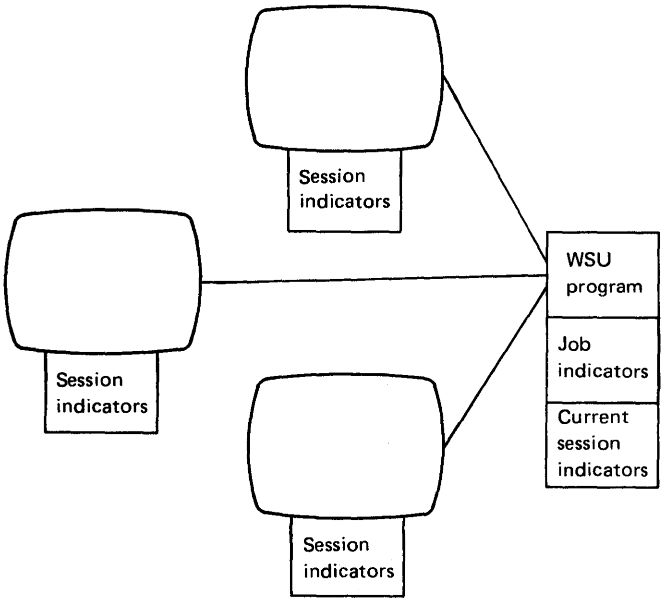
For example, if indicator JA is set on by a display station, that indicator is on for all other display stations that use the program.

Session Indicators

The session indicators are:

AE	(accept sequence error) Allows operators to bypass required displays by means of the Accept-Sequence-Error command key.
ES	(end-of-sequence-set processing level) Indicates the end of the primary display sequence.
EW	(end-of-work-session processing level) Indicates the end of a work session.
IW	(work-session-initiation processing level) Indicates the start of a work session.
RC	(recovery of work session) Indicates when an operator resumes a work session that either ended abnormally or had records removed from the chain. The WSU program must have a transaction file in order for this indicator to be used.
RS	(resume work session) Indicates when an operator resumes a work session that ended normally. The WSU program must have a transaction file in order for this indicator to be used.
SA-SN, SP-SY	(session level) Provide mode-to-mode saving of indicator settings.
U1-U8	(external) Provide a way of passing indicator settings from the OCL that starts the WSU program to a session.

As the following shows, WSU maintains a separate copy of these indicators for each display station using the same program. These indicators retain their settings when a display station changes from one operating mode to another.



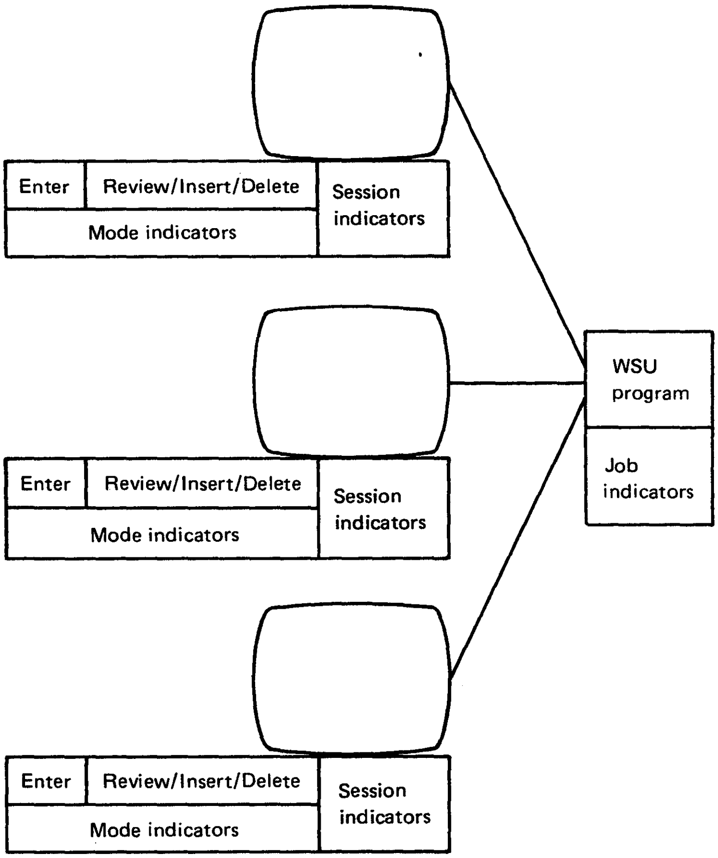
For example, if indicator SA is set on by a display station, the indicator is on only for that display station. Other display stations that use the program might have indicator SA set off or set on.

Mode Indicators

The mode indicators are:

AC	(accept command key) Signals the condition of any current user-defined command key request.
CG	(current group) Indicates that the transaction file record selected by the operator for review is either the most recently entered header record or a data record that follows the most recently entered header record.
DL	(delete record) Indicates that the transaction file record selected by the operator for review is a record that is to be logically deleted when the processing cycle ends.
IN	(insert mode) Indicates the insert operating mode.
IP	(input to process) Signals operations which should be performed only when there could be some data fields input from the display.
IS	(initiate transaction sequence) Signals the start of a new function request.
KG-KL, KQ-KY	(command key) Indicate which user-defined command key an operator presses.
PG	(program mode) Temporarily prevents operator-initiated changes in transaction file content or display sequence.
RP	(repeated display) Indicates that a display has reappeared because of an MSG operation.
RR	(review record) Signals an on-going review or delete function request.
RU	(rover update) Determines if the PUT operation with no record identifying indicator can be successfully run for the transaction file.
RV	(review mode) Indicates the review operating mode. This indicator is also on during insert mode.
01-89	Provide general indicator use in a program; for example, indicate when operations should occur, indicate results of operations, and specify field attributes for a display. Indicators 90 through 99 are reserved for WSU and cannot be used in your WSU programs.

As the following shows, WSU maintains a separate copy of these indicators for enter mode and review/insert/delete mode.



For example, if indicator 10 is set on by a display station in enter mode, that indicator is on only for that mode. When the operating mode changes, the indicator settings for enter mode are saved, and the mode indicators are cleared and adjusted for review mode.

The indicators for enter mode are restored when enter mode resumes. The mode indicators are cleared and adjusted whenever review mode is reset (for example, repeated use of the Roll Up and Roll Down keys causes review mode to be reset for each press of a key).

Figure 3-1 is a summary of the job, session, and mode indicators.

Initial Settings of Indicators

For the first operator that calls a WSU-generated procedure, all indicators are off except IJ, IW, any external indicators (U1 through U8) that have been set on, and RS and/or RC if operators had previously run the program and used the transaction file.

For operators that call that same procedure later, all indicators are off except for the IW indicator, any external indicators that have been set on, any job indicators that have been set on by other display stations that have called the procedure, and the RS and/or RC indicators.

	WSU Controlled	**Programmer Controlled**
Job	EJ	EJ
	IJ	JA-JN, JP-JY
Session	ES	AE
	EW	ES
	IW	EW
	RC	RC
	RS	RS
		SA-SN, SP-SY
		U1-U8
Mode	AC	AC
	CG (off during enter mode)	IP
	DL (off during enter mode)	IS
	IN (off during enter mode)	KG-KL, KQ-KY
	IP	PG
	KG-KL, KQ-KY	RP
	PG	01-89
	RP	
	RR	
	RU	
	RV (off during enter mode)	
	IS	

Figure 3-1. Indicator Summary

The following explains each of these indicators.

AC (Accept Command Key) Indicator

Set on by: WSU, when processing for a display is started again with new user command key settings, that is, with one and only one of the user command key indicators (KG-KL, KQ-KY) set on.

Set off by:

- WSU, when the command key indicators are cleared
- User, with a SETOF operation, to signal that no command key request is still waiting.

This indicator may be left on to condition later operations which should or should not be run depending on what request has just been handled. When the AC indicator is set off, the program can skip code which checks each command key individually.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

AE (Accept Sequence Error) Indicator

Set on by:

- The SETON, MSG, or IMSG operation
- Results of an arithmetic or compare operation.

Set off by:

- The SETOF operation
- Results of an arithmetic or compare operation.

When on, this indicator allows operators to bypass sequence errors by pressing the Accept-Sequence-Error command key (Cmd 13). (A sequence error occurs when an operator tries to bypass a required display or ES processing.) When off, this indicator causes an error message when operators try to bypass a required display or ES processing.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

CG (Current Group) Indicator

Set on by: WSU, when a transaction file record that has been selected by an operator for review is either the most recently entered header record or a data record that follows the most recently entered header record.

Set off by:

- WSU, when a transaction file record that has been selected by an operator for review is from a different group than the most recently entered record.
- WSU, when an operator switches from review mode to enter mode.

Where coded: Conditioning Indicators on C-specifications.

DL (Delete Mode) Indicator

Set on by: WSU (along with the RV indicator), when an operator selects delete mode. Refer to *Review Mode* in Chapter 2 for an explanation of how an operator can select delete mode.

Set off by: WSU, when an operator returns to enter mode (by means of the Resume Entry command key); when enter mode resumes as a result of ES, EJ, or EW being set on; or when an operator selects a record to review.

Where coded: Conditioning Indicators on C-specifications.

EJ (End of Job Processing Level) Indicator

Set on by:

- WSU, when both of the following conditions exist:
  - EW processing is completed when only one operator is using the program and that operator has specified end of work session on the WSU display
  - The program is a never-ending program (NEP) and the operator has not specified restart on the WSU display.
- WSU, when any abnormal-ending condition occurs
- A SETON, MSG, or IMSG operation
- Results of an arithmetic or compare operation.

Set off by:

- WSU, when all operations conditioned by the EJ indicators complete
- Results of an arithmetic or compare operation
- A SETOF operation.

Where coded:

- Format ID on an S-specification
- Processing Function, Conditioning Indicators, and Resulting Indicators on C-specifications.

ES (End of Sequence Set Processing Level) Indicator

The ES indicator can be used to cause operations to be performed at the end of a sequence of displays. Typically, these operations might be used to perform clean up functions needed before later operations or to inform the operator (by means of the IMSG operation) that the display sequence has been completed and the next display will be the first of the sequence set.

Set on by:

- WSU, each time, except the first, when the first display in the primary sequence is next to appear. ES does not turn on for secondary display sequences
- WSU, when the EW or EJ indicator turns on (except during ES processing)
- A SETON, MSG, or IMSG operation
- Results of an arithmetic or compare operation.

Set off by:

- WSU, when all operations conditioned by the ES indicator are complete and neither the EW indicator nor the EJ indicator is on
- Results of an arithmetic or compare operation
- A SETOF operation.

Where coded: Processing Function, Conditioning Indicators, and Resulting Indicators on C-specifications.

Setting the ES indicator on when no ES processing is present in a WSU program causes WSU to select the first display in the primary sequence. If present in the program, ES processing is performed even if the primary sequence is not started before the session ends.

EW (End of Work Session Processing Level) Indicator

Set on by:

- WSU, when an operator specifies EW on the WSU display or selects a session
- WSU, when the EJ indicator turns on
- A SETON, MSG, or IMSG operation
- Results of an arithmetic or compare operation.

Set off by:

- Results of an arithmetic or compare operation
- A SETOF operation.

Note: The SETOF operation cannot be used to cancel a previous end session or end job request.

Where coded:

- Format ID on an S-specification
- Processing Function, Conditioning Indicators, and Resulting Indicators on C-specifications.

Note: If the EW indicator is used to force an end of session for an operator at a display station with an ID that does not match the work session ID, that operator is allowed to restart by selecting a different session.

IJ (Job Initiation Processing Level) Indicator

Set on by: WSU, when the first operator starts the WSU program.

Set off by:

- WSU, when all operations conditioned by the IJ indicator are complete
- WSU, when the next display in a different processing function is displayed (normally, this is the IW display).

Where coded:

- Format ID on an S-specification
- Processing Function and Conditioning Indicators on C-specifications.

IN (Insert Mode) Indicator

Set on by: WSU, when an operator selects insert mode by means of the Insert Record command key.

Set off by: WSU, when the mode switches to enter or review.

Where coded: Conditioning Indicators on C-specifications.

IP (Input to Process) Indicator

Set on by: WSU, when a user response is accepted and input data has been returned to the program.

Set off by:

- WSU, when the next user response (PUTS/MSG/IMSG) is accepted
- User, with a SETOF operation.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

IS (Initiate Transaction Sequence) Indicator

Set on by:

- WSU, when a display is automatically selected to start a program defined sequence for a new request such as:
  - Selection of the first display for a session. Sign on at a work station or session selection from the WSU display is considered to be a request to begin entering records
  - Selection of the first display for a review or review/delete request
  - Selection of the first display for a begin insert request.

Set off by:

- User, with a SETOF operation:
  - To condition operations which should be done, but done only once as mode is set or reset
  - To begin data entry
  - To begin review of a record selected by the operator
  - To begin inserting records after a record being reviewed by the operator.

The user may assume that a transaction file record has been read when the IS indicator is set on in review mode to signal the start of a review functions sequence. However, the user must provide his own loop control for initializing fields when adding a chain of records for a started entry or insert function sequence, or when using GET/PUT operations (with no record ID) in a record updating loop inside a WSU sequence.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

IW (Work Session Initiation Processing Level) Indicator

Set on by:

- WSU, after all operations conditioned by the IJ indicator are completed (for only the first operator that begins the WSU program)
- WSU, when any operator after the first operator begins the WSU program.

Set off by:

- WSU, when all operations conditioned by the IW indicator are completed
- WSU, when the next display in a different processing level is displayed (normally, this is the first display in the primary sequence).

Where coded:

- Format ID on an S-specification
- Processing Function and Conditioning Indicators on C-specifications.

JA-JN, JP-JY (Job) Indicators

Set on by:

- A SETON, MSG, or IMSG operation
- Results of an arithmetic or compare operation.

Set off by:

- Results of an arithmetic or compare operation
- A SETOF operation.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

KG-KL, KQ-KY (Command Key) Indicators

The KG-KL and KQ-KY indicators are user command keys; WSU uses command keys 1 through 6 and 13 through 15.

Set on by:

- A SETON, MSG, or IMSG operation.
- Results of an arithmetic or compare operation.
- WSU, when an operator presses the corresponding command key. WSU turns all other command key indicators off. The command keys and indicators are:
  - KG: Command key 7
  - KH: Command key 8
  - KI: Command key 9
  - KJ: Command key 10
  - KK: Command key 11
  - KL: Command key 12
  - KQ: Command key 16
  - KR: Command key 17
  - KS: Command key 18
  - KT: Command key 19
  - KU: Command key 20
  - KV: Command key 21
  - KW: Command key 22
  - KX: Command key 23
  - KY: Command key 24

Set off by:

- WSU, when a display is shown that has Reset Keyboard specified (Also, WSU sets all command key indicators off when a display is shown by means of the IMSG or MSG operation).
- Results of an arithmetic or compare operation.
- A SETOF operation.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

PG (Program Mode) Indicator

Set on by: WSU, when an operator tries to bypass the current display or to select a record to review/delete/insert.

Set off by:

- WSU, when the processing for a display is ended (no PUTS operation run to select the next display and no C-specification to run for the current display)
- WSU, when an operator EW indicator or session selection is accepted from the WSU display
- User, when the **lock** provided by the PG indicator for the display sequence or the transaction file is no longer required.

When the PG indicator is on, an error message is issued if the operator tries to bypass the current display or to select a record to review/delete/insert. When the PG indicator is on, the operator can still request functions by means of enabled user-defined command keys or WSU keys 1, 3, or 13, and can still use the WSU display. The PG indicator can be used to condition operations.

Where coded: Conditioning Indicators on C-specifications.

RC (Recovery of Work Session) Indicator

Set on by:

- WSU, when an operator resumes a work session that had been abnormally ended
- WSU, when the transaction file might contain records which are removed from the chain for the session.

Set off by: A SETOF operation.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications. In order to use this indicator, a WSU program must have a transaction file.

Note: Once on, this indicator is always on for the session unless a SETOF operation turns the indicator off. The indicator turns on each time the session is restarted until the WSU Recover procedure or the WSU Extract procedure is used to produce a transaction file that has no records which have been removed from the chain for the work session.

RP (Repeated Display) Indicator

Set on by: WSU, when a MSG operation causes a display to reappear.

Set off by:

- WSU, when a new display is selected in one of the following ways:
  - WSU selects an appropriate display because:
    - Display processing ends normally.
    - You bypass a display using the Bypass Display command key (Cmd 2).
    - You select an alternative function (RV, DL, or IN).
  - You select a nonhelp display from the WSU display. (When the RV indicator is on, you can select the current display again.)
  - A PUTS operation selects:
    - A different keyboard-resetting display. (A help display resets the keyboard).
    - A display again without processing.
- User, when coding loops containing both MSG operations, for which WSU sets on the RP indicator, and preprocessing PUTS operations which should be conditioned on NRP.

Where coded: Conditioning Indicators on C-specifications.

Notes:

1. The RP indicator setting is not changed when processing for a current display resumes at the C-specification entry that follows a preprocessing PUTS operation.
2. Because RP is a mode level indicator, the enter mode setting is restored when enter mode is resumed.

RR (Review Record) Indicator

Set on by: WSU, after reading and retrieving data fields from a transaction file record selected for review by the operator.

Set off by: WSU, when the IN indicator is set on.

The RR indicator can be used in conjunction with the IS indicator to condition operations which should be done to prepare for displaying or updating an operator selected record or to adjust for a new current review record.

The RR indicator can be regarded as a signal that performing of a PUT operation coded with a record type identifier will result in the replacement of the current review record. Any GET-before-PUT-required operation is done automatically by WSU.

The RR indicator can also be used as a signal that the current relative record number can be found in *RLRR instead of *RLNO.

Where coded: Conditioning Indicators on C-specifications.

RS (Resume Work Session) Indicator

Set on by: WSU, when an operator resumes a work session.

Set off by: A SETOF operation.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications. In order to use this indicator, a WSU program must have a transaction file.

Note: Once on, this indicator is always on for a session until a SETOF operation turns the indicator off.

RU (Rover Update) Indicator

Set on by: WSU, when *RLRU is updated after a successful GET operation.

Set off by: WSU, when a GETNR/GETNH/GETPH operation is attempted, a review request is accepted, enter mode is resumed, or a transaction file PUT operation with no record type indicator is attempted.

RU may be used to condition running of a PUT operation which could cause the program to have an abnormal end with a GET-before-PUT-required error code.

Where coded: Conditioning Indicators on C-specifications.

RV (Review Mode) Indicator

Set on by: WSU, when an operator selects review or delete mode. Refer to Chapter 2 for an explanation of how an operator can select review mode. The RV indicator remains on during insert mode.

Set off by: WSU, when an operator returns to enter mode (by means of the Resume Entry command key), or when enter mode resumes as a result of ES, EJ, or EW being set on.

Where coded: Conditioning Indicators on C-specifications.

SA-SN, SP-SY (Session Level) Indicators

Set on by:

- A SETON, MSG, or IMSG operation.
- Results of an operation.

Set off by:

- A SETOF operation.
- Results of an arithmetic or compare operation.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

U1 through U8 (External) Indicators

U1 through U8 are indicators accessed by each requesting display station at the initiation of a work session and saved by WSU at the normal end of a work session. Because the indicators are saved, they can be used to pass information from session to session. These indicators are not saved if the work session ends abnormally.

Set on by:

- A SWITCH OCL statement entered by an operator or entered as part of a procedure which is not an MRT procedure.
- A SETON, MSG, or IMSG operation.
- Results of an arithmetic or compare operation.

Set off by:

- A SWITCH OCL statement entered by an operator or performed from a procedure which is not an MRT procedure (see note).
- Results of an arithmetic or compare operation.
- A SETOF operation.
- The WSU Extract, WSU Create, or WSU Recover procedure when that procedure begins running.

Where coded: Conditioning Indicators and Resulting Indicators on C-specifications.

Note: The procedure that WSU generates to call a WSU program is an MRT procedure. When an MRT procedure is first requested, the SSP copies that requester's external switches into an area associated with the MRT procedure. OCL statements within the MRT procedure access this copy. A SWITCH OCL statement within the WSU-generated procedure does not change the external switch setting associated with the first requester, but instead changes the copy of those switches.

Changes made to the external switches from within the program affect the requesting display station's external switches rather than the copy of the switches.

01 through 89 Indicators

Set on by:

- WSU, when the indicator is used to identify a record type and when that type of record is read from the file.
- WSU, when the indicator has been coded in columns 49 and 50 (Not-Found Indicator) of the T-specification or an M-specification, and when a record to be read cannot be found in the file.
- WSU, when an error occurs for a GETNR, GETNH, GETPR, GETPH, GET, PUT, or PUTN operation and the indicator has been coded in columns 61 and 62 (Error Indicator) of the T-specification or M-specification.
- A SETON, MSG, or IMSG operation.
- Results of an arithmetic or compare operation.

Set off by:

- WSU, when the indicator is used to identify a record type and when a different type of record is read from the same file.
- Results of an arithmetic or compare operation.
- A SETOF operation.
- WSU, each time a record is selected for review.

Where coded:

- Record Identifying Indicator on I-specifications.
- Not-Found Indicator, Header Record Identifying Indicator, and Error Indicator on the T-specification.
- Not-Found Indicator and Error Indicator on M-specifications.
- Format ID, Sound Alarm, Null Fill, Blink Cursor, Review Mode, and Insert Mode Record Identifying Indicators on S-specifications.
- Output Data, Position Cursor, Protect Field, High Intensity, Blink Field, Nondisplay, Reverse Image, and Underline on D-specifications.
- Conditioning Indicators and Resulting Indicators on C-specifications.

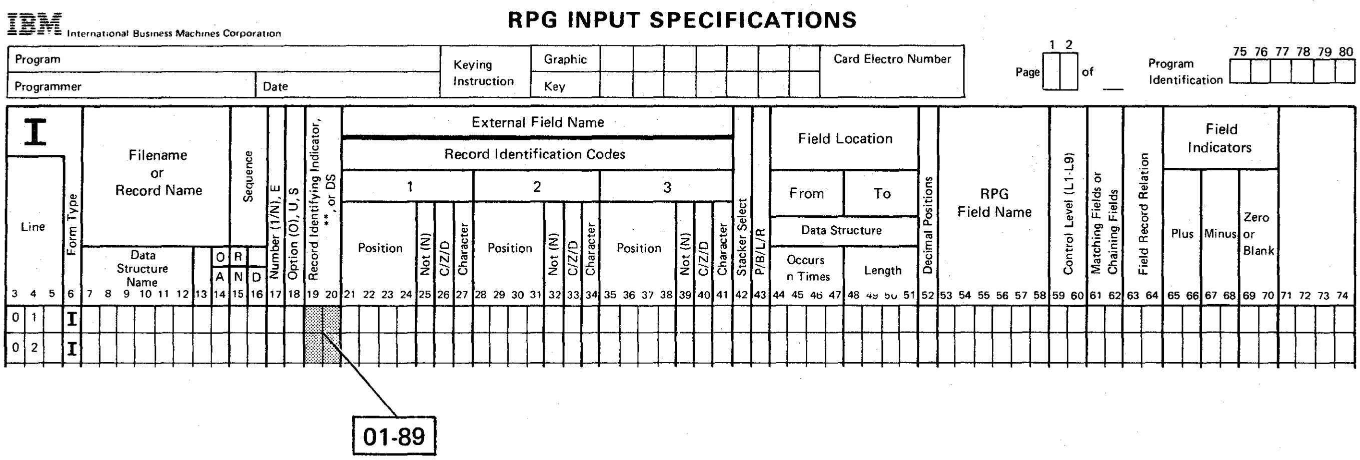
Notes:

1. WSU does not automatically place a record identification code into a record when you write the record with a PUT operation. If you want to write a code in each record, define a field for the code on the I-specification, then move a value into this field.

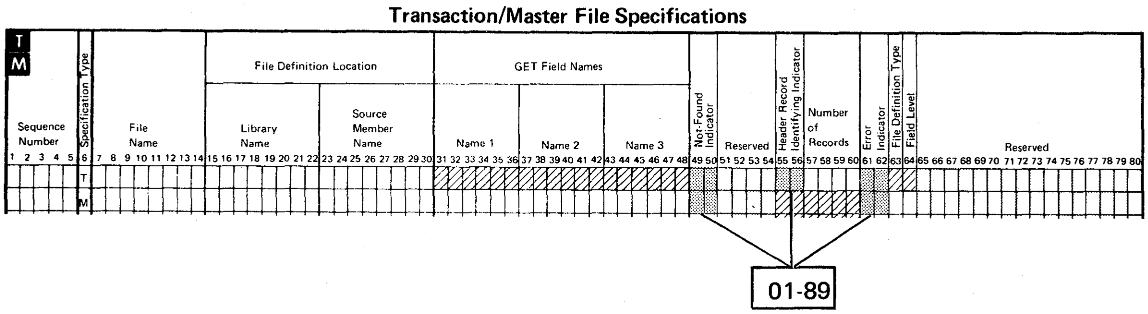
2. AND/OR and Record Identification Codes should be blank if there is only one record type in the file, or if you want all record types processed the same way.

3. If the program is to be used with the WSU Create procedure (WSUTXCR) to build a transaction file, blank records must not satisfy any of the record types; otherwise, blank records in your file will be tagged as data records, and your file will be full before you start.

Figure 3-2 summarizes where to code indicators.



WSU turns this indicator off when a read is attempted from the file.
WSU turns this indicator on when the record type is read from the file.



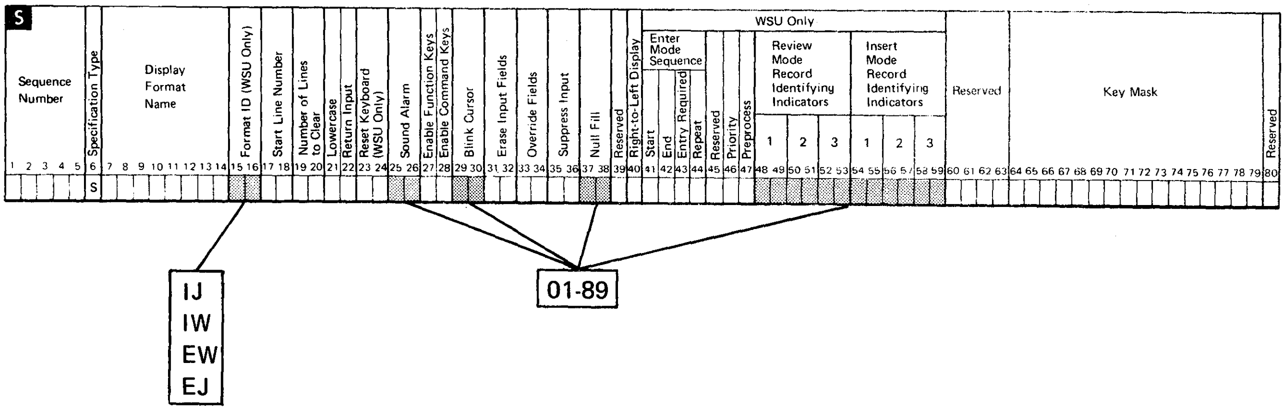
Columns 49 and 50: WSU turns the indicator on when a record cannot be read or a record has been read but not identified.

Columns 55 and 56: The indicator must identify a record type in the transaction file. This indicator allows operators to page from header record to header record when reviewing records.

Columns 61 and 62: This indicator turns off each time a GET operation is done for the file, and turns on if an error occurs in trying to read from the file, or if a record-not-found condition occurs and a not-found indicator has been omitted from the T- or M-specification, and from the C-specification.

Figure 3-2 (Part 1 of 3). Where to Code Indicators

Display Control Specification

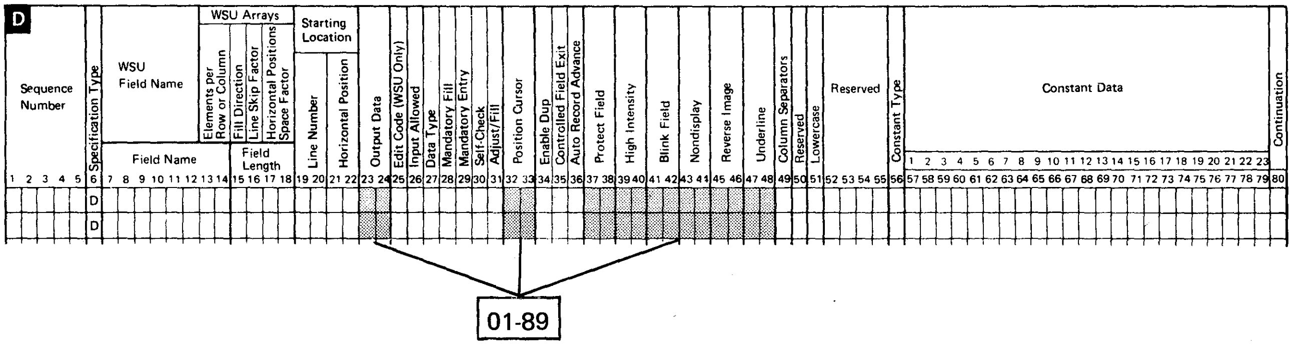


Columns 15 and 16: These indicators cause displays to automatically appear for a processing level.

Columns 25 and 26, 29 and 30, 37 and 38: These indicators control display attributes.

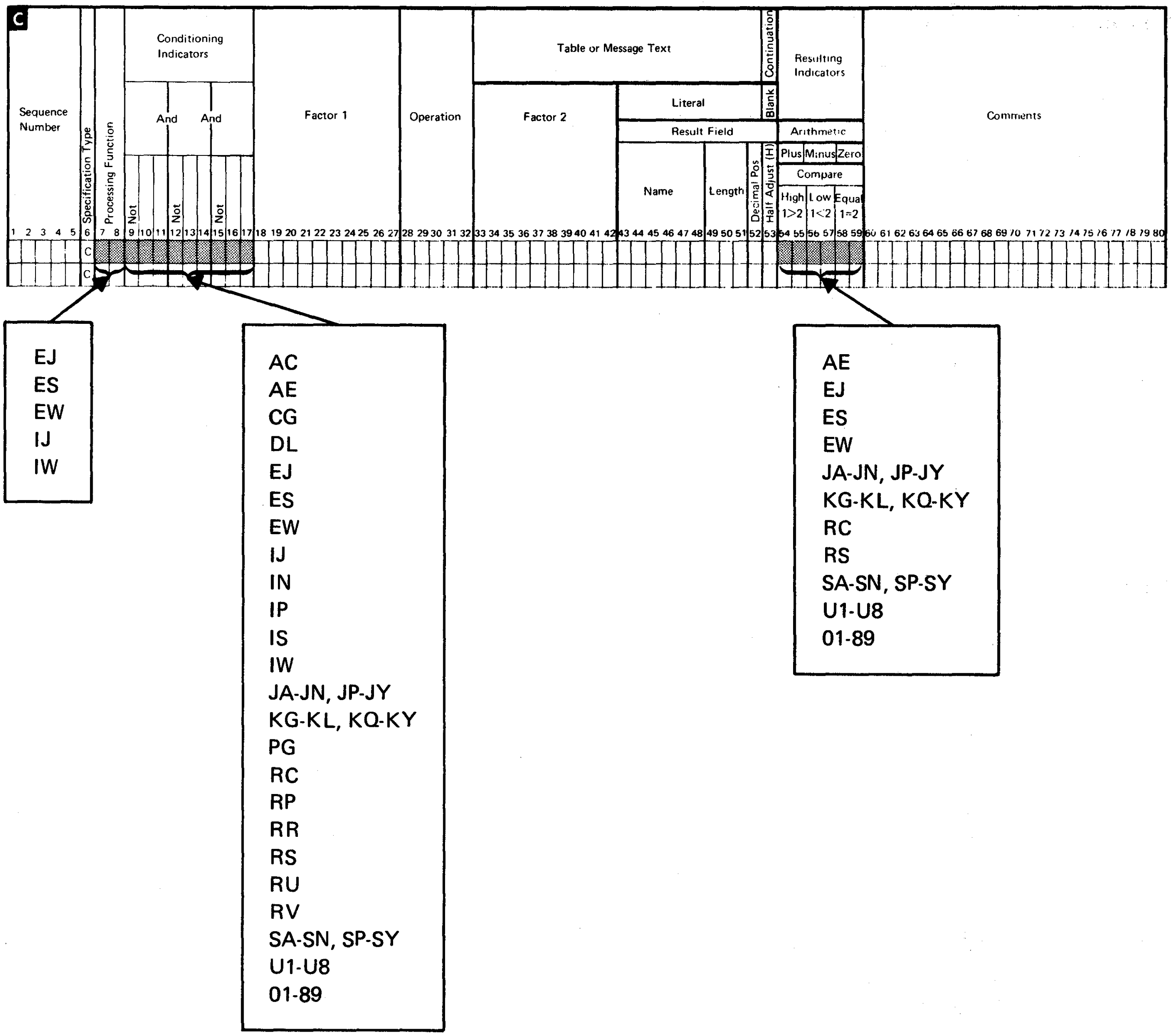
Columns 48 through 59: These indicators allow control of how operators use review and insert modes and must match record identifying indicators in the transaction file.

Field Definition Specification



These indicators control attributes of data on the display.

Figure 3-2 (Part 2 of 3). Where to Code Indicators



EJ
ES
EW
IJ
IW

AC
AE
CG
DL
EJ
ES
EW
IJ
IN
IP
IS
IW
JA-JN, JP-JY
KG-KL, KQ-KY
PG
RC
RP
RR
RS
RU
RV
SA-SN, SP-SY
U1-U8
01-89

AE
EJ
ES
EW
JA-JN, JP-JY
KG-KL, KQ-KY
RC
RS
SA-SN, SP-SY
U1-U8
01-89

Columns 7 and 8:	These indicators cause operations to occur for certain processing levels.
Columns 10 through 17:	These indicators condition operations.
Columns 54 through 59:	Figure 17-5 in Chapter 17 shows how these indicators can be used with WSU operations.

Figure 3-2 (Part 3 of 3). Where to Code Indicators

Chapter 4. Coding Files

This chapter provides information about:

- Transaction and master files: Collections of information used by WSU.
- File processing utilities: Utilities used to process WSU transaction files and non-WSU files.
- Trailers: 13 bytes of information used by WSU to chain records.
- File definition coding: F- and I-specifications used for coding WSU programs.
- File coding: WSU J-, T-, and M-specifications used for coding WSU programs.

WSU Files

Master Files

A WSU program can read from and write to as many as 20 master files.

With Distributed Data Management (DDM), WSU programs on the local system may use master files that are on a remote system. Refer to Appendix B for a brief description of DDM.

Master files can be direct files (in which records are assigned specific record positions) or indexed files (in which the position of the record is recorded in an index).

WSU supports alternative indexes for direct and indexed master files. Coding WSU for an alternative indexed master file is exactly the same as for any other indexed master file. Remember, if the index contains duplicate keys, WSU will only access the first occurrence of the key. To create the alternative indexes, you must use the BLDINDEX procedure. Refer to the *S/36 System Reference Manual,*SC21-9020, for an explanation. For more information on alternative indexes, see the *Concepts and Programmer's Guide*, SC21-9019.

WSU also supports non-contiguous keys for alternative indexed master files. Coding WSU for a non-contiguous keyed master file is exactly the same as for other alternative indexed files except for the F-specification. Refer to Chapter 12 for an explanation. To specify non-contiguous keys when creating an alternative indexed file, you must use parameters on the BLDINDEX procedure. Refer to the *S/36 System Reference Manual*, SC21-9020, for an explanation.

WSU requires an F-specification and I-specifications that describe each master file. Refer to *File Definition Coding* in this chapter for information on coding master files.

The F-specification indicates the physical characteristics of the file (for example, record length and key length) as well as the type of organization (indexed or direct). The I-specifications describe the type and size of fields within the file as well as the type(s) of records in the file.

You can code the following operations to read records from and write records to a master file:

- GET (for reading records)
- PUT (for changing records)
- PUTN (for adding records to a master file).

Refer to Chapter 18, *Operation Codes*, for a description of these and other operations.

Sharing Master Files

A master file can be shared by two or more WSU programs. *Modifying a WSU-Generated Procedure* in Chapter 8 describes how you indicate a shared master file.

A WSU program that updates a shared master file should not have any of the following operations coded between reading a record (GET) and updating that record (PUT):

- PUTS
- MSG
- IMSG
- GET from another file.

This is because protection is removed from the original record read when one of these operations occurs. Another program might read and update the record before this program updates it. Operations are covered in Chapters 7 and 17.

A master file can be changed by two or more display stations using the same WSU program. In this situation, WSU automatically protects the records so that they are updated correctly.

Transaction Files

One of the main functions of a WSU program is to allow records to be added to or changed in a transaction file. WSU manages the transaction file automatically so that records can be added or changed easily. The transaction file is always a direct file, and its records are separated *logically* according to the work session for which they were entered. WSU protects the file so that records entered for one work session cannot be read or modified from another work session.

Not all WSU programs use transaction files, but when used, the transaction file should be unique to each WSU program.

Distributed Data Management (DDM) lets WSU use a remote transaction file.

WSU requires an F-(file description) specification and I-(input) specifications that describe the transaction file. Refer to *File Definition Coding* in this chapter for ways that these specifications can be stored. This chapter also contains an explanation of the required entries to the F- and I-specifications.

The transaction file is a direct file to WSU, regardless of your entries on the F-specifications. The format of records written to the transaction file is defined on the I-specifications.

At first, the transaction file contains blank records. The first record in the file is a job-control record. WSU creates this record before creating the first session chain. The first record in each session chain is a work-session-control record. See *Trailers* in this chapter for a discussion of transaction file trailers. As operators enter data, WSU replaces blank records with data. As it writes a record to the file, WSU reserves an area for the next record in the chain (either the chain of work-session-control records or the chain of header and data records for a work session). The last record entered in the chain points to the area where the next record will be written.

WSU maintains a reserved field, *RLNO, that keeps track of records in the transaction file. *RLNO contains the relative record number of the next record to be written to the transaction file. You can reference this field in your program if you need the information that it contains. If the transaction file is not defined in a WSU program, *RLNO is zero during the program's running. Refer to Chapter 3, *Reserved Fields and Indicators*, for more information on *RLNO.

Writing Records to and Reading Records from the Transaction File

WSU does not automatically write records to the transaction file. You must code operations to write to and read from the transaction file. These operations (described in Chapter 18) are:

Operation	**Explanation**
PUT	Adds a record to the transaction file or changes the record just read.
GETPR	Reads the previous record from the transaction file.
GETPH	Reads the previous header record from the transaction file.
GETNR	Reads the next record from the transaction file.
GETNH	Reads the next header record from the transaction file.

Notes:

1. Next record and previous record in the descriptions of the operations are relative to the last record that was read for the session. GET operations in a program are not always related to the record on the display.

2. WSU will not read beyond the beginning or end of the chain of records for the current work session. Instead, a record-not-found condition occurs.

3. If a transaction file is not defined in a WSU program, the GETPR, GETPH, GETNR, and GETNH operations cause terminal errors during program generation.

Removing Trailers and Control Records from the Transaction File

One way to remove trailers and control records from a transaction file is to use the WSU Extract (WSUTXEX) procedure. This procedure is described in Chapter 19, *File Processing Procedures*.

Another way to remove trailers and control records is to use SUBR22, a subroutine that is provided by RPG II. Refer to the manual, *Programming with RPG II*, for a description of SUBR22.

Recovering a Transaction File

The WSU Recover procedure (WSUTXRV) can be used to recover a transaction file. This recovery is usually necessary after a WSU session or WSU program ends abnormally. The procedure can be used to drop problem-causing records from the file or to recover records that were lost when the abnormal end occurred. This procedure is described in Chapter 19.

File Processing Procedures

WSU supplies procedures with which you can perform the following processing activities on transaction files:

- With files:

  - Prepare a WSU transaction file for use by a non-WSU program WSUTXEX procedures or RPG SUBR22 subroutine)
  - Create a WSU file from a non-WSU file (WSUTXCR procedure)
  - Concatenate several transaction files (WSUTXEX, WSUTXCR procedures)
  - Recover a WSU transaction file (WSUTXRV procedure).

- With records:

  - Remove blank records (WSUTXEX procedure)
  - Create one record chain from several record chains (WSUTXEX, WSUTXCR procedures)
  - Reorder or exclude record chains (WSUTXEX, WSUTXCR procedures)
  - Reclaim partially inserted records (WSUTXRV procedure)
  - Remove partially inserted and logically deleted records (WSUTXRV procedure or WSUTXEX, WSUTXCR procedures)
  - Print relative record numbers (WSUTXEX procedure).

Refer to Chapter 19, *File Processing Procedures*, for a further discussion of the WSU procedures.

Typical Uses of the WSU Procedures

The following paragraphs describe some typical uses of the WSU procedures.

Prepare a WSU Transaction File for Use by a Non-WSU Program

You can prepare a WSU transaction file for use by a non-WSU program in any of the following ways:

- Use the WSUTXEX procedure to extract records in logical order from the WSU transaction file.

- Use the WSUTXEX procedure to extract only data records from the WSU transaction file.

- Use the WSUTXEX procedure to copy the WSU transaction file and remove trailers from its records.

Remove Blank Records

The WSUTXEX procedure can extract and print nonblank records from a WSU transaction file. This function is useful for removing the blank records that WSU leaves in its transaction files and increasing the number of available records in the file.

Create One Record Chain from Multiple Record Chains

The WSUTXEX procedure and the WSUTXCR procedure can extract one or more record chains, create a WSU transaction file from them, and assign the same display station ID to all of the records in the new file. WSU only allows a display station to review records for the session that is currently active for the display station. Therefore, these procedures are useful when you need an efficient way to view multiple chains from one display station.

To temporarily change a transaction file to assign the same display station ID to all the records requires that the file provide extra space for a new trailer for the extracted records. The display station ID is assigned in the new trailer. Because of the longer record length, a different WSU program is required to review the records. The extra space can be removed from the records when the changed transaction file is no longer needed, which results in a file that can be processed (or created again for processing) by the original WSU program.

Print Relative Record Numbers

The WSUTXEX procedure can select and print records for a work session in logical order. The relative number of each record is printed. The WSU display allows an operator to review a record by entering its relative record number.

Create a WSU File from a Non-WSU File

The WSUTXCR procedure can create a WSU transaction file from a non-WSU file. This function is useful if the initial data entry is done using a non-WSU program and you want to convert the file to a WSU transaction file so that it can be maintained interactively by a WSU program.

Reorder or Exclude Record Chains

The logical order of the session chains in a WSU file is the order in which the work-session-control records are in the file.
The WSUTXEX and WSUTXCR procedures are useful for putting the session chains in another order (for example, by display station ID W1, W2, . . .) or for excluding unwanted chains. The WSUTXEX procedure can extract session chains in a specified order (collecting the records into a single file) and the WSUTXCR procedure can create a WSU transaction file from them.

Concatenate Multiple Transaction Files

The WSUTXEX procedure can collect records from multiple WSU transaction files into a single file and from them the WSUTXCR procedure can create one WSU transaction file on the disk. A WSU program can create multiple files, for example, if it is run on more than one System/36 or if it is run on different days on the same System/36. The WSUTXEX and WSUTXCR procedures are useful to put the data back into one file.

Reclaim or Remove Partially Inserted Records

The WSUTXRV procedure can reclaim or remove partially inserted records from a WSU transaction file. Records that were being inserted when the WSU program ended abnormally become partially inserted because the pointers in their trailers are not correct. These records can cause errors when WSU tries to reuse them as blank chain records. The WSUTXRV procedure is useful for reclaiming or removing those records before they confuse an operator.

Remove Partially Inserted and Logically Deleted Records

The WSUTXRV procedure or the WSUTXEX procedure in conjunction with the WSUTXCR procedure can be used to delete partially inserted and logically deleted (unchained) records from a WSU transaction file. Partially inserted and logically deleted records should be removed from a WSU file to improve program performance. When there are no partially inserted or logically deleted records in a file, WSU can access a record that is requested by the relative record number if the record has the current work session ID in the trailer. Otherwise, WSU must scan the session chain for the selected record to ensure that access can be allowed.

Recover a WSU Transaction File

The WSUTXRV procedure can recover a WSU transaction file. A transaction file in need of recovery can produce the following symptoms:

- The WSU program that uses it cannot be initialized.
- A work session cannot be resumed.
- A new work session cannot begin.
- Data records cannot be added from a current work session.
- The job or a session ends abnormally when an unidentified record is encountered.

Trailers

All data records and control records in a WSU transaction file contain 13 bytes of trailer information at the end of each record. Trailers contain information that WSU uses to chain records together for each distinct work session. WSU allows logical insertion and deletion of records within each chain and prevents more than one display station at a time from accessing records that are entered for a session.

Figure 4-1 shows the logical organization of the transaction file. Because several display stations can enter input to a program at the same time, records from the various work sessions become mixed in the file. WSU uses the last 13 bytes of each record for control information (the trailer). Trailers allow WSU to access the transaction file as separate chains of records, one chain per work session. The record length on the F-specification that describes the transaction file must include 13 bytes for the trailer.

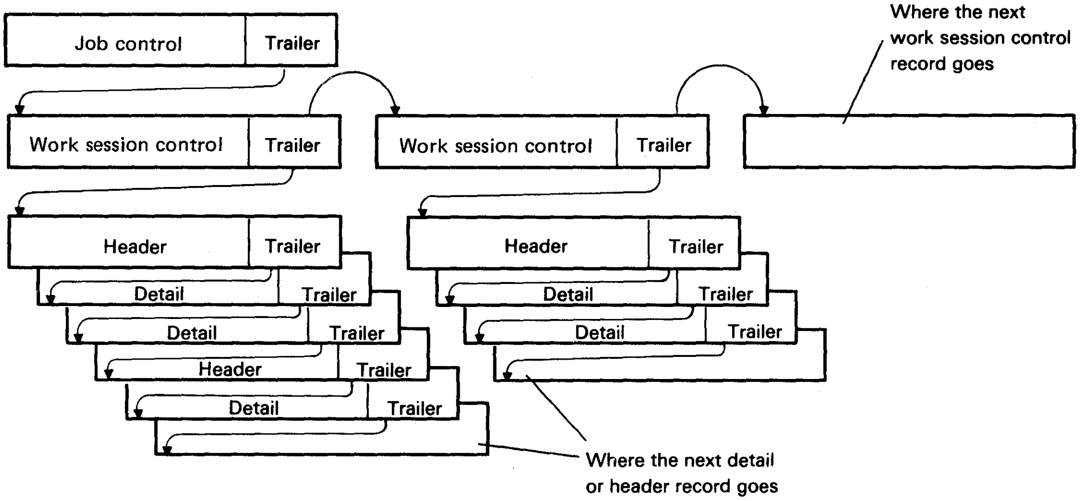
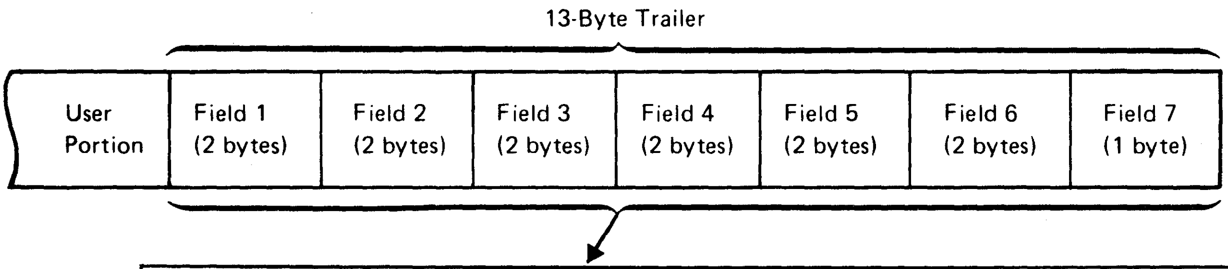


Figure 4-1. Logical Organization of the Transaction File

Each display station has a 2-character symbolic identifier (such as W1). When an operator runs a WSU program, this identifier is used as a work session identifier and is included in the trailer of each record added to the transaction file during the work session. If the display station is authorized (by the WSU program) to allow the operator to select a different work session after the WSU program has started, the work session identifier of records added to the transaction file can be different than the work station identifier. This is because the work session identifier in the trailer always matches the identifier of the work session the operator is using.

Figure 4-2 shows the contents of the 13-byte trailers in the transaction file's job control, work session control, and data records. All pointers in the transaction file are 2-byte, binary relative record numbers (relative to zero). If the transaction file is read in an RPG II program without using the RPG II subroutine SUBR22, add one to each pointer to obtain an actual relative record number.



Job Control Record	User Portion:	Not used (hex 40s)
	Field 1:	Largest relative record number allocated in the file
	Field 2:	Point (relative record number) to the first work session control record (hex 0001)
	Field 3:	Not used (hex 0000)
	Field 4:	Not used (hex 0000)
	Field 5:	Not used (hex 0000)
	Field 6:	Not used (hex 0000)
	Field 7:	Hex F1: Previous execution of this job ended normally Hex F2: Previous execution of this job ended abnormally or the job is running Hex F3: No job control record present. This file is a new file.
Work Session Control Record	User Portion:	Not used (hex 40s)
	Field 1:	Largest relative record number allocated in this work session chain
	Field 2:	Pointer to the next work session control record
	Field 3:	Pointer to the last logical header record in this chain
	Field 4:	Pointer to the last logical record in this chain
	Field 5:	Pointer to the first logical data record in the work session chain
	Field 6:	Identifier for this work session
	Field 7:	Hex F3: Work session ended normally Hex F4: Work session ended abnormally or work session is running Hex F5: Work session was restarted after ending abnormally. The file might contain unchained records entered for the session. Deleting a record changes it to F3. Hex 40: End of work session control record chain
Data Record	User Portion:	User data
	Field 1:	Hex 0000: Record can be reclaimed if it is not chained to the previous record but it can be validly chained to the previous record Hex FF00: Record cannot be reclaimed
	Field 2:	Not used (hex 0000)
	Field 3:	Pointer to the previous header record. Field 3 is hex 0000 for the first record in the file or for files that have no header records
	Field 4:	Pointer to the previous record in the work session chain. Field 4 is hex 0000 for the first data record in a chain
	Field 5:	Pointer to the next record in a work session chain. For the last record, field 5 points to a blank record
	Field 6:	Display station identifier for this work session
	Field 7:	Hex F6: This record is a detail data record Hex F7: This record is a header data record Hex 40: This is the last data record in the chain of data records.

Figure 4-2. Trailer Contents

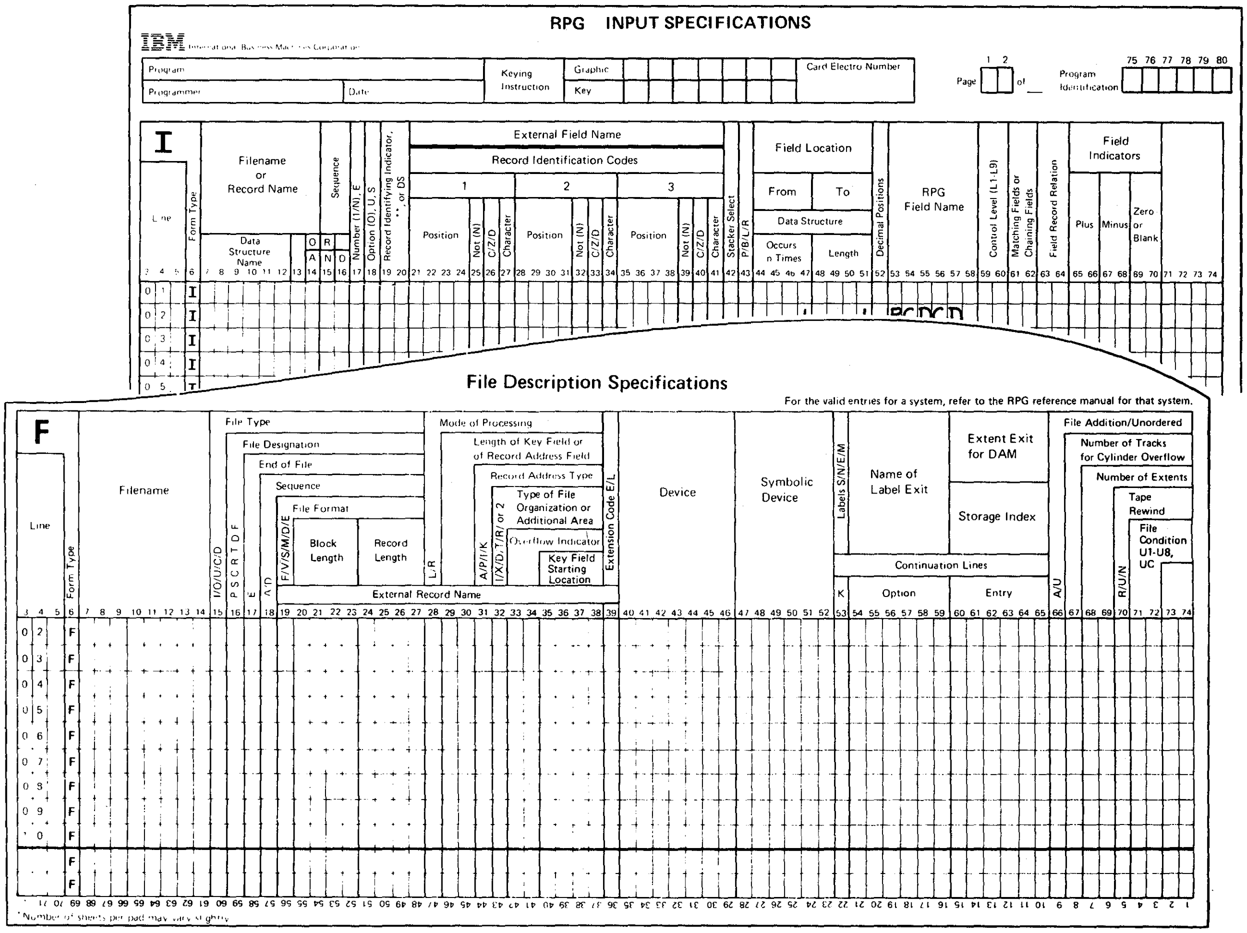
File Definition Coding

A WSU program must contain a file definition which contains descriptions of the transaction file and master files used by a WSU program. This file definition contains file description and input specifications. Figure 4-3 shows the F- and I-specification coding forms. The information needed to run your WSU program is taken by WSU from the RPG F- and I-specification sheet; the information that is not needed is ignored.

The F-specification contains the specifications that describe the transaction and master files. The I-specification contains the specifications for the following fields that are used in your program:

- Transaction-file fields
- Master-file fields
- Local-data-area fields
- Session-level fields.

Examples of coded F- and I-specifications are in this chapter and in Chapter 11, *WSU Example Programs*, and Chapter 12, *F-(File Description) and I-(Input) Specifications.*



RPG INPUT SPECIFICATIONS

Program
Programmer
Date
Keying Instruction
Graphic
Key
Card Electro Number
Page of
Program Identification

Filename or Record Name
Data Structure Name
Sequence
Number (1/N), E
Option (O), U, S
Record Identifying Indicator, **, or DS
External Field Name
Record Identification Codes
Position
Not (N)
C/Z/D
Character
Stacker Select
P/B/L/R
Field Location
From
To
Data Structure
Occurs n Times
Length
Decimal Positions
RPG Field Name
Control Level (L1-L9)
Matching Fields or Chaining Fields
Field Record Relation
Field Indicators
Plus
Minus
Zero or Blank

File Description Specifications

For the valid entries for a system, refer to the RPG reference manual for that system.

Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
Block Length
Record Length
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
Extension Code E/L
External Record Name
Device
Symbolic Device
Labels S/N/E/M
Name of Label Exit
Extent Exit for DAM
Storage Index
Continuation Lines
K
Option
Entry
A/U
File Addition/Unordered
Number of Tracks for Cylinder Overflow
Number of Extents
Tape Rewind
File Condition U1-U8, UC
R/U/N

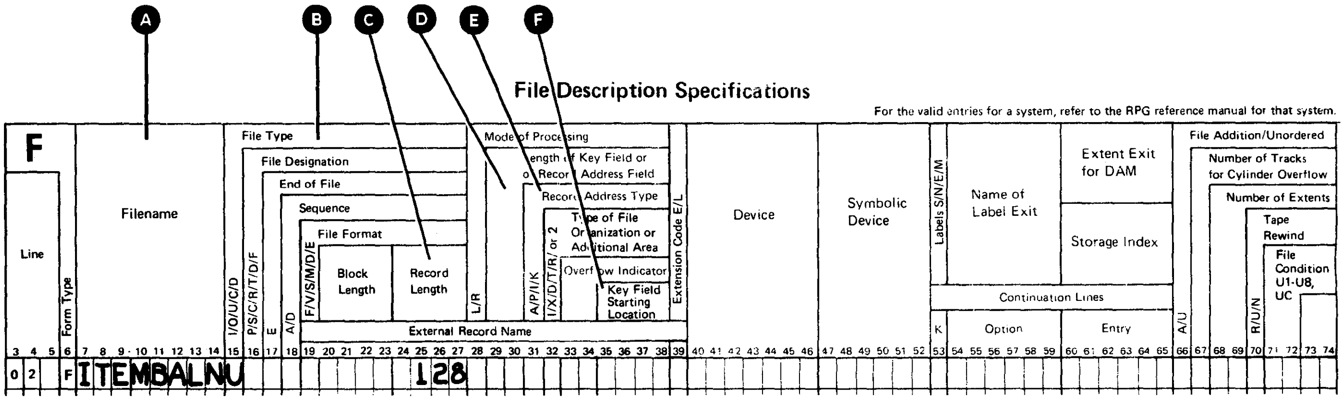
*Number of sheets per pad may vary slightly

Figure 4-3. File Definition Coding Forms

F-Specification

The following entries are required for both transaction and master files.

- The *File Name* (columns 7 through 14) must be entered in columns 7 through 14 to specify the name of the transaction or master file used. Ⓐ
- The *File Type* (column 15) must be either an I (input) or an U (update); you can use either if your F-specification is used only in connection with a WSU program. Ⓑ
- The *Record Length* (columns 24 through 27) must include the length of the transaction record plus 13 bytes used by WSU to maintain the records in the transaction file by display station. Ⓒ



The following entries are required only for indexed master files.

- The *Length of Key Field* (columns 29 and 30) specifies the length in bytes of the record key. Ⓓ
- The *Record Address Type* (column 31) specifies either a packed decimal record key or an alphameric record key. Ⓔ
- The *Key Field Starting Location* (columns 35 through 38) specifies the starting position of the key in each record of an indexed master file or contains EXTK for a non-contiguous keyed file. Ⓕ

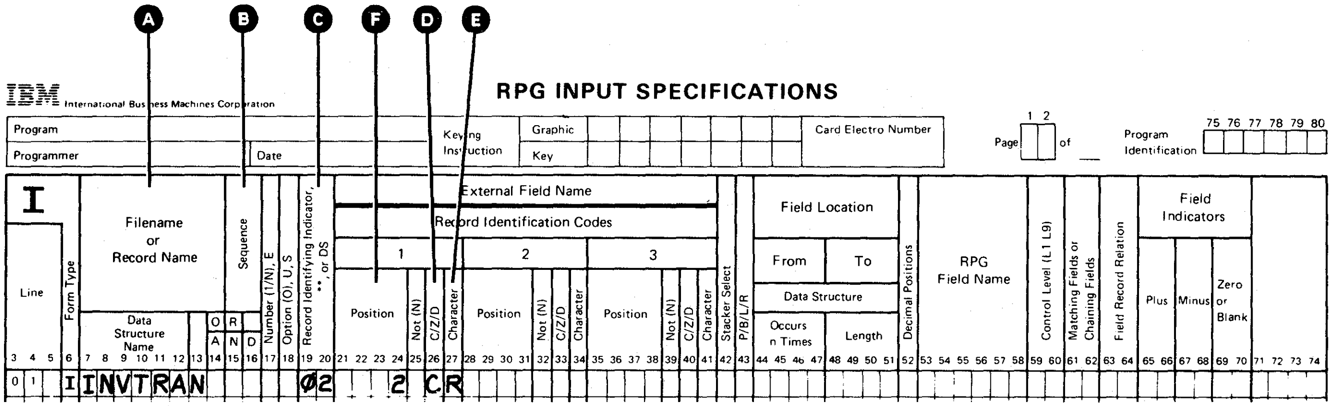
I-Specification

A WSU program's file definition must contain I-(input) specifications for the transaction-file fields, master-file fields, local-data-area fields, and session-level fields that are used in the program.

An I-specification can have two types of lines: record lines and field lines. Record lines use columns 7 through 42; columns 43 through 58 must be blank. There can be a maximum of 255 record lines. Field lines use columns 43 through 58; columns 7 through 42 must be blank.

Record lines include:

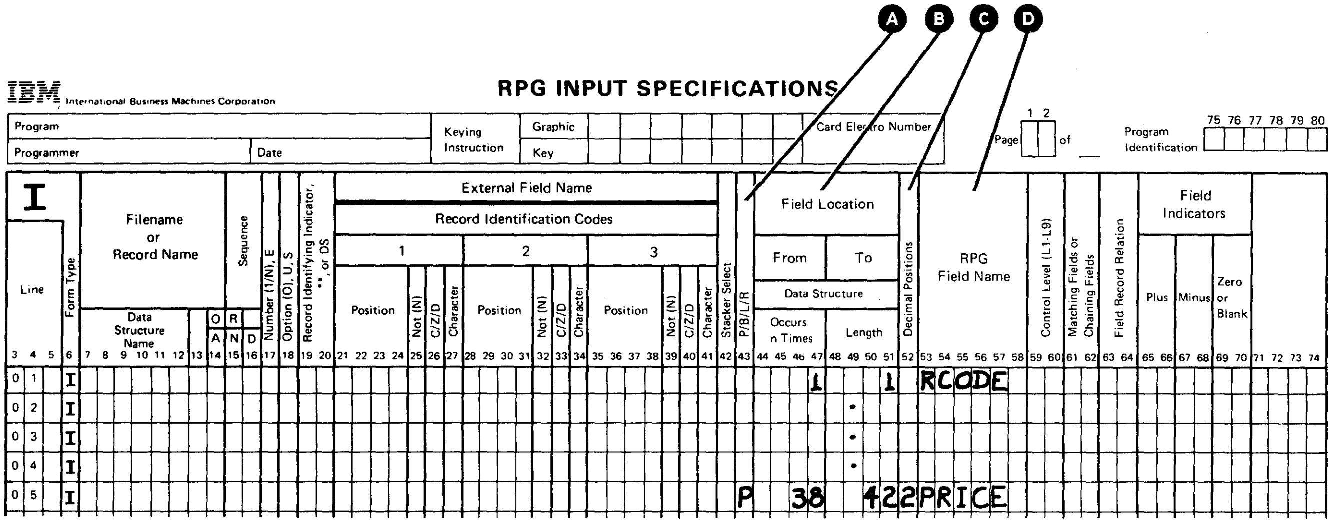
- *File Name* (columns 7 through 14) which specifies the name for a transaction file, master file, group of session-level fields, or local data area. Ⓐ
- *AND/OR* (columns 14 through 16) which specifies that you need more than three record identification codes or if either one of the codes can be present to identify the record. AND/OR is discussed later in this chapter. Ⓑ
- *Record Identifying Indicator* (columns 19 and 20) which specifies the record type. Ⓒ
- *Record Identification Codes* which include:
  - *Not (N)* (columns 25, 32, 39) specifies that the following C/Z/D of the record identification character must not be in the *Position* location. (columns 21 through 24, 28 through 31, 35 through 38)
  - *C/Z/D* (columns 26, 33, 40) specifies which portion of the *Character* (columns 27, 34, 41) is used as the record identification character. Ⓓ Ⓔ
  - *Position* (columns 21 through 24, 28 through 31, 35 through 38) specifies the location in the record of the record identification character. Ⓕ



0 1 I INVTRAN 02 2 C R

Field lines include:

- *P/B/L/R* (column 43) which specifies packed decimal, binary, zoned decimal, or an alphameric field. Ⓐ
- *Field Location* (columns 44 through 55) which specifies the beginning and ending locations of the field. Ⓑ
- *Decimal Positions* (column 52) which specifies the number of decimal positions to the right of the decimal point in this numeric field. Ⓒ
- *Field Name* (columns 53 through 58) which specifies the name of this field. Ⓓ



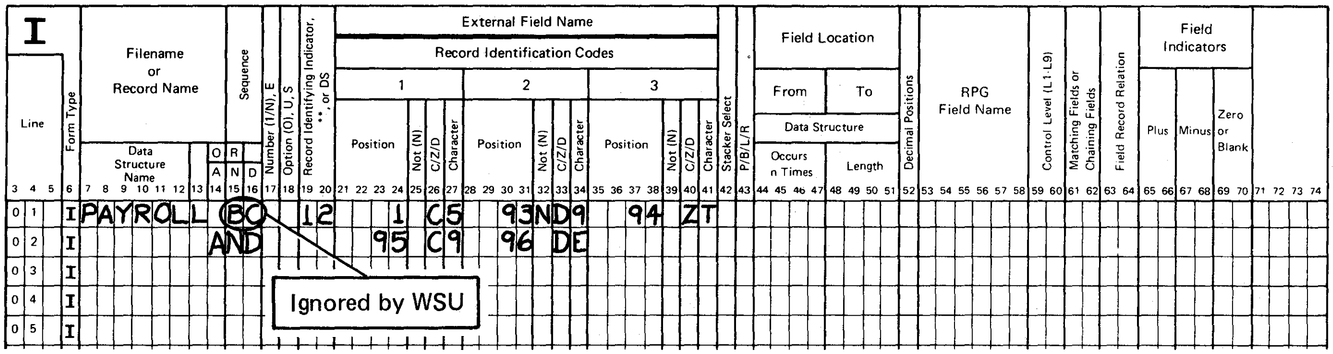
```
0 1 I                                               1      1 RCODE
0 2 I                                                  .
0 3 I                                                  .
0 4 I                                                  .
0 5 I                                        P    38     422PRICE
```

AND/OR Relationships

AND Relationship: You can code a maximum of three record identification characters on one specification line. If the identification code consists of more than three characters, you can use an AND line. This means that the first three identification characters are described on the first line. Additional identification characters are described on following lines by coding AND in columns 14 through 16 to indicate the continued lines.

Figure 4-4 shows record identification codes consisting of five characters. The first character is in position 1, the other four characters are in positions 93, 94, 95, and 96. Since you can code only three identifying characters on one line, the word AND on the next line indicates that the last two characters of the code are part of the previous record identification entries.

You can code a maximum of 20 AND lines to describe the record identification codes for a record. The record must contain all the characters indicated as record identification characters before the record identifying indicator turns on.



01 I PAYROLL BC 12 1 C5 93ND9 94 ZT
02 I AND 95 C9 96 DE
03 I
04 I
05 I

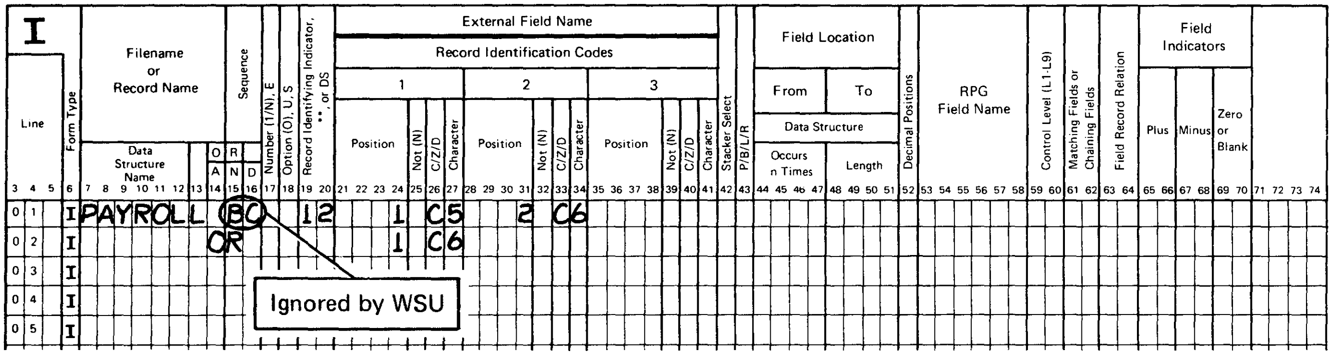
Ignored by WSU

Figure 4-4. Example of AND Line

OR Relationship: Two different record types can be identified by different characters in the same physical position in the field. You can code OR lines to indicate which of the codes can be present to identify the record. You can code a maximum of 20 OR lines for each record. Also, you can use OR lines to specify a second type of record that has different record identification codes, but the same fields.

Figure 4-5 shows the use of an OR line to describe record identification codes. The record is identified by either of two different codes: a code consisting of a 5 in position 1 and a 6 in position 2, or a code consisting of a 6 in position 1.

Note: If AND lines and OR lines are combined, the total number of such lines for one record type cannot exceed 20.



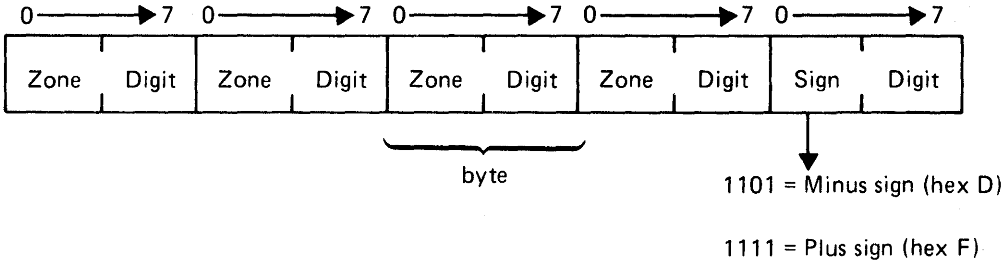
```
  01 IPAYROLL BC  12   1 C5   2 C6
  02 I       OR        1 C6
  03 I
  04 I
  05 I
```

Figure 4-5. Example of OR Line

Data Formats

System/36 supports zoned decimal, packed decimal, binary, and alphameric data formats. The data format is specified in column 43 of the I-specification.

Zoned Decimal Format (Blank): In zoned decimal format each byte of storage contains 1 decimal digit. In the zoned decimal format, each byte of storage is divided into a 4-bit zone portion and a 4-bit digit portion. The zoned decimal format looks like this:

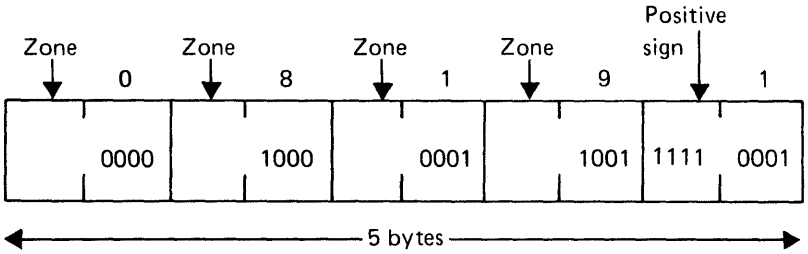


Note: WSU does not perform data verification on numeric data. The value of the digit portion of a character is assumed to be the numeric value of that character.

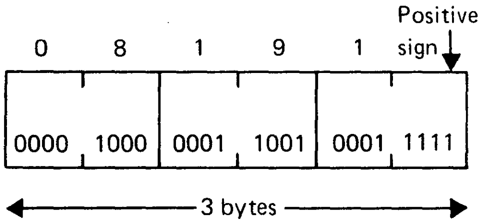
The zone portion of the low-order byte indicates whether the decimal number is positive or negative. In zoned decimal format, each digit in a decimal number includes a zone portion; however, only the low-order zone portion serves as the sign. Figure 4-6 shows the zoned decimal format of a number in storage.

Once data is read into the computer, it must be represented in the zoned decimal format before it can be used. Thus, data can be stored on disk and read into the computer in the zoned decimal format, thereby eliminating the need to convert the field. However, storing numeric data (decimal numbers) on disk in either the packed decimal or the binary format provides more efficient use of disk storage space.

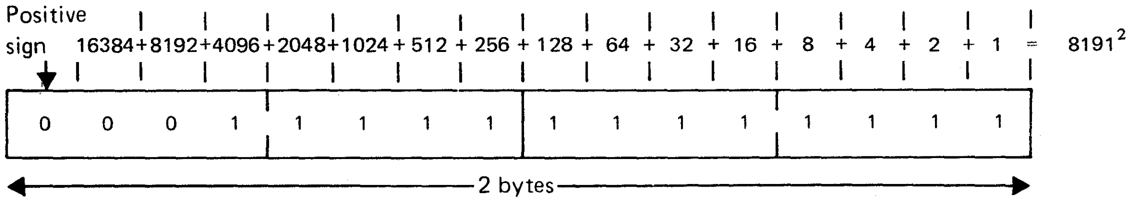
Zoned Decimal Format:[1]



Packed Decimal Format:



Binary Format:



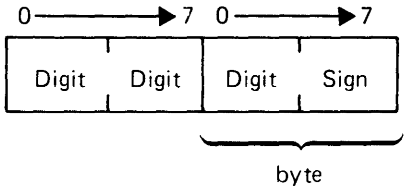
[1] If 8191 is read into storage as a zoned decimal field, it occupies 4 bytes of the 5 bytes shown. However, if it is converted to packed decimal format, it occupies 3 bytes; then when it is converted back to zoned decimal format, it occupies 5 bytes.

[2] To obtain the numeric value of a positive binary number, add the value of the bits that are on (1); the sign bit is not included. To obtain the numeric value of a negative binary number, add the values of the bits that are off (0) plus on; the sign bit is not included.

Figure 4-6. Zoned Decimal, Packed Decimal, and Binary Representation of 8191

Packed Decimal Format (P): In packed decimal format each byte of disk storage (except for the low-order byte) contains 2 decimal digits. Because many of the fields in a disk file contain decimal digits, you can save disk space by storing these fields in the packed decimal format.

In the packed decimal format, each byte of disk storage, except the low-order byte, is divided into two 4-bit digit portions. The rightmost portion of the low-order byte contains the sign (plus or minus) for that field. The packed decimal format looks like this:



The sign portion of the low-order byte indicates whether the numeric value represented in the digit portions is positive or negative. In the packed decimal format, the sign is included for each decimal number; however, the zone portion is not given for each digit in the number. Compare how the decimal number 8191 is represented in packed decimal format with its zoned decimal representation shown in Figure 4-6.

Because data must be represented in zoned decimal format to be processed by the computer, you must give the WSU program an indication when input fields are in another format. Entering a P in column 43 indicates that the input field is in the packed decimal format and that the system must convert this field to the required zoned decimal format.

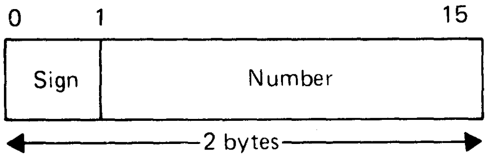
When a packed decimal field is converted to a zoned decimal field, the zoned decimal field always contains an odd number of bytes. If a zoned decimal field with an even number of bytes is converted to a packed decimal field and then converted back to a zoned decimal field, the resulting zoned decimal field also contains an odd number of bytes.

Packed fields can be up to 8 bytes long. The following chart shows the packed equivalents for zoned decimal fields up to 15 bytes long:

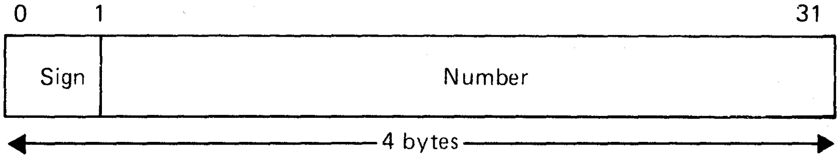
Zoned Decimal Length in Bytes	**Packed Length in Bytes**
15	
14	8
13	
12	7
11	
10	6
9	
8	5
7	
6	4
5	
4	3
3	
2	2

Binary Format (B): The binary format allows you to save even more disk storage space than you can save using the packed decimal format. In the binary format, each field on disk must be either 2 or 4 bytes long.

Each 2-byte binary field consists of a 1-bit sign followed by a 15-bit numeric value. In binary format, a decimal number as high as 9999 requires only 2 bytes of disk storage. For each 2-byte binary field stored on disk, WSU automatically sets aside 4 bytes of storage for the field when it is converted to zoned decimal format. A 2-byte field in binary format looks like this:



Each 4-byte binary field consists of a 1-bit sign followed by a 31-bit numeric value. In binary format, a decimal number as high as 999,999,999 requires only 4 bytes of disk storage. For each 4-byte binary field stored on disk, WSU automatically sets aside 9 bytes of storage for the field when it is converted to a zoned decimal format. A 4-byte field in binary format looks like this:



In each case, the sign portion of the high-order byte indicates whether the numeric value is positive (sign bit off) or negative (sign bit on). Positive numbers are represented in true binary notation with a zero sign bit. Negative numbers are represented in two's complement notation with a one sign bit. (The two's complement of a number is formed by taking the true binary notation, changing all 1's to 0's and all 0's to 1's, and adding 1 to the new number). The bits between the sign position and the leftmost significant bit of the integer are always the same as the sign bit. When the number is positive, these bits are zeros; when the number is negative, all these bits are ones.

Notice that, in the binary format, the zone position of the decimal number is not given. Compare how the decimal number 8191 is represented in binary format with packed and zoned decimal representation (see Figure 4-6).

Because data must be represented in zoned decimal format to be processed by the computer, you must give the WSU program an indication when input fields are in another format. Entering a B in column 43 indicates that the input field is in the binary format and that the system must convert this field to the required zoned decimal format.

Examples of Coding F- and I-Specifications

Figure 4-7 shows an example of coding F- and I-specifications for a transaction file.

This F-specification defines a transaction file named TRANS that has a 525-byte record length (512 bytes plus 13 bytes for each trailer). Remember that WSU ignores entries on the F-specification that it does not need.

Regardless of the F-specification entries, WSU creates the transaction file as a direct output file. In order to use the transaction file as described by an F-specification for an indexed file, an RPG II program must change the file to an indexed file.

These I-specifications further define the transaction file. TRANS has two record types: 01 and 02.

The 01 record has all of the following characteristics:

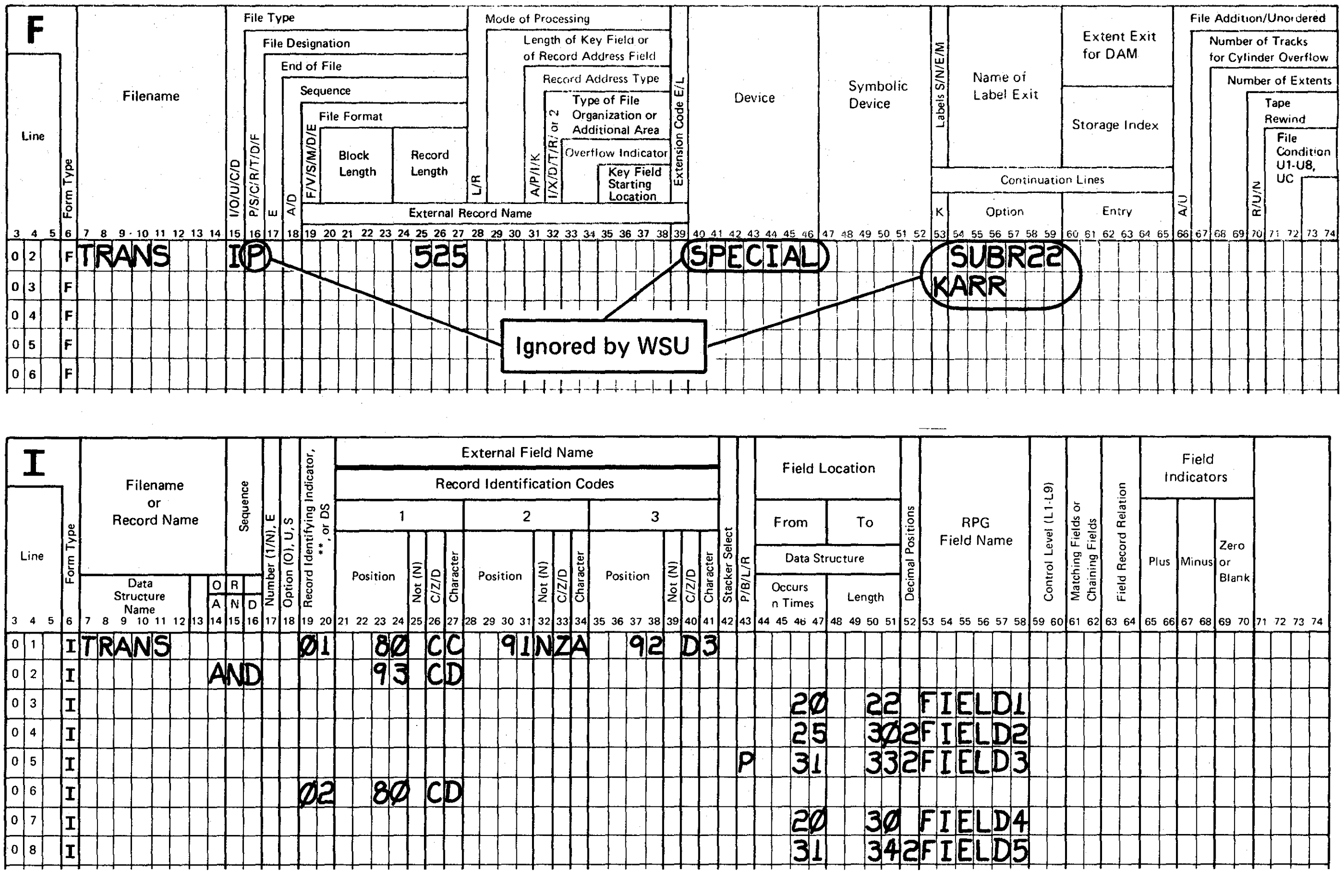
- Character C in position 80
- The absence of the following characters in position 91:
  - &
  - A through I (Not zone A eliminates A through I because the zone character for A through I is the same.)
- One of the following characters in position 92:
  - C
  - L
  - T
  - 3
- Character D in position 93.

Note: Refer to the manual *Programming with RPG II* for information about hexadecimal representation of characters.

Record 01 has three fields: FIELD1 is alphameric and begins in position 20; FIELD2 is numeric with two decimal positions and begins in position 25; and FIELD3 is packed decimal numeric with two decimal positions and begins in position 31.

Record 02 is identified by the character D in position 80. It has two fields: FIELD4 is an alphameric field that begins in position 20, and FIELD5 is a numeric field with two decimal positions that begins in position 31.

Figure 4-8 shows an example of coding F- and I-specifications for a direct master file. Figure 4-9 shows an example of these specifications for an indexed master file.

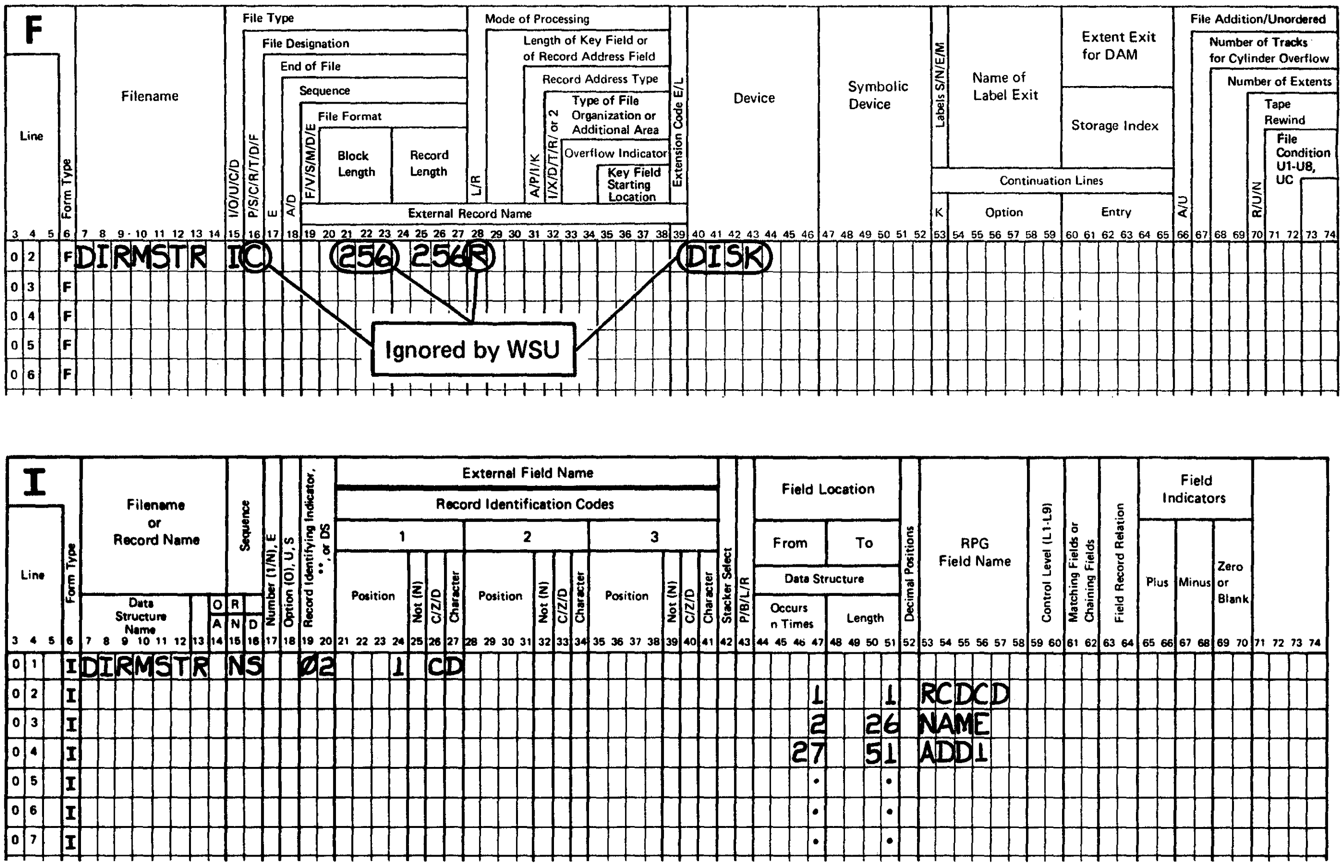


Ignored by WSU

```
  02 FTRANS   IP       525            SPECIAL       SUBR22
  03 F                                              KARR
```

```
  01 ITRANS       01  80 CC  91NZA  92 D3
  02 I       AND      93 CD
  03 I                                       20  22 FIELD1
  04 I                                       25  302FIELD2
  05 I                                    P  31  332FIELD3
  06 I            02  80 CD
  07 I                                       20  30 FIELD4
  08 I                                       31  342FIELD5
```

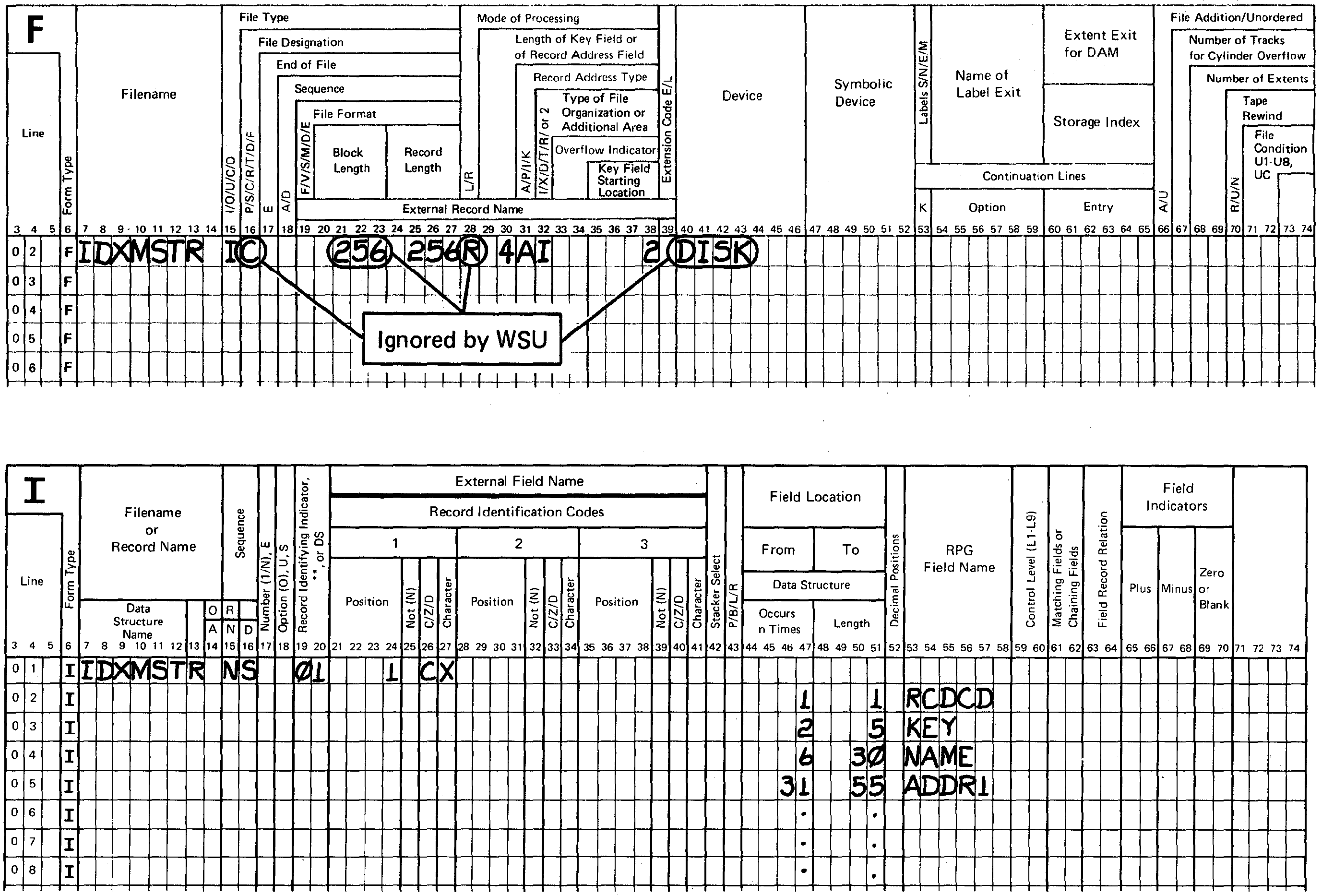
Figure 4-7. Example F-Specifications and I-Specifications



```
02 FDIRMSTR IC    256 256R           DISK
03 F
04 F
05 F
06 F
```

```
01 IDIRMSTR NS  02   1 CD
02 I                                        1   1 RCDCD
03 I                                        2  26 NAME
04 I                                       27  51 ADD1
05 I
06 I
07 I
```

Figure 4-8. Example F- and I-Specifications for a Direct Master File



```
02 FIDXMSTR IC    256 256R 4AI     2 DISK
03 F
04 F
05 F
06 F
```

```
01 IIDXMSTR NS  01   1 CX
02 I                                        1   1 RCDCD
03 I                                        2   5 KEY
04 I                                        6  30 NAME
05 I                                       31  55 ADDR1
06 I
07 I
08 I
```

Figure 4-9. Example F- and I-Specifications for an Indexed Master File

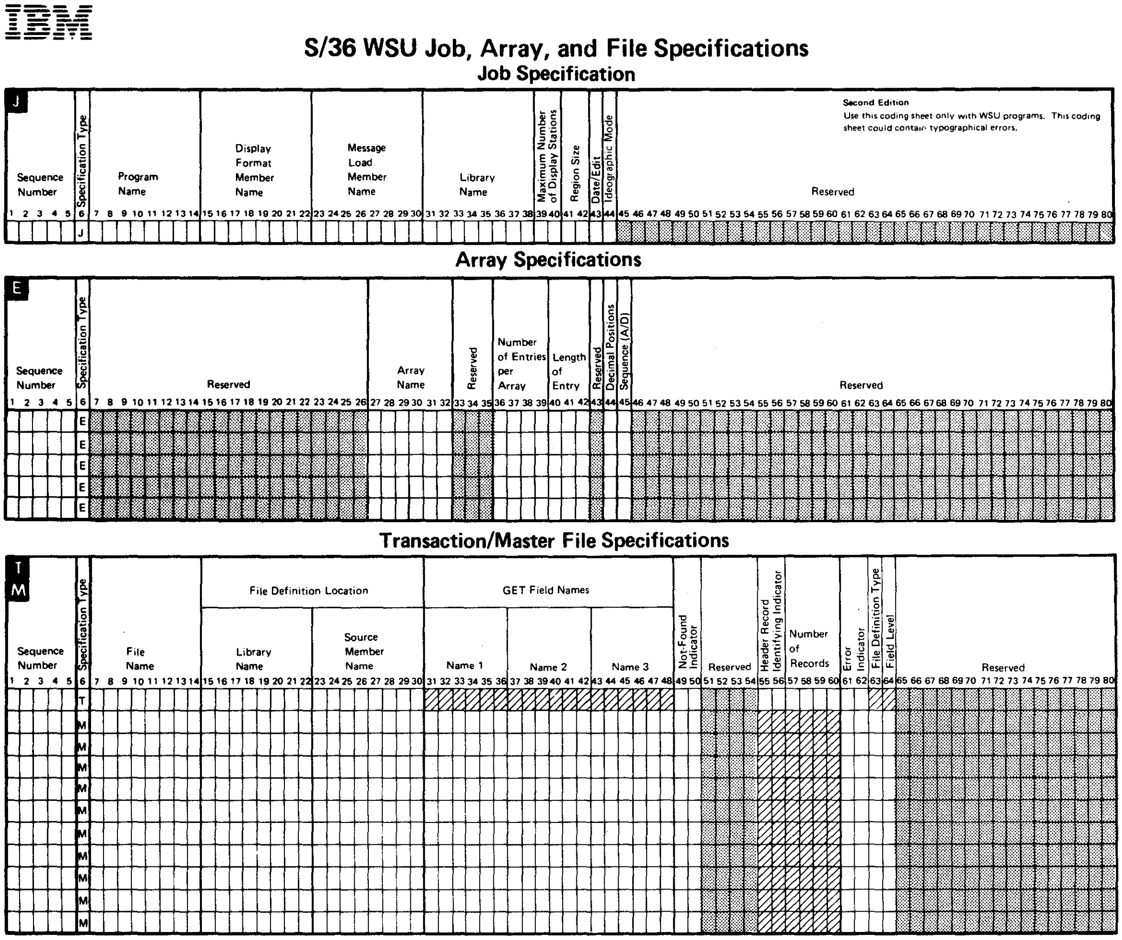
File Coding

When coding your source program you must include:

- J-specification (one per program)
- T-specification (if used, one per program)
- M-specification (if used, as many as needed for your program).

Note: For WSU data entry programs, you **must** have either a transaction file or a master file.

These specifications can all be coded on one form: the WSU Job, Array, and File Specifications coding form (see Figure 4-10). These specifications must come before the rest of your source program (the S-, D-, and C-specifications).



IBM

S/36 WSU Job, Array, and File Specifications

Job Specification

Second Edition
Use this coding sheet only with WSU programs. This coding sheet could contain typographical errors.

Sequence Number 1 2 3 4 5	Specification Type 6	Program Name 7 8 9 10 11 12 13 14	Display Format Member Name 15 16 17 18 19 20 21 22	Message Load Member Name 23 24 25 26 27 28 29 30	Library Name 31 32 33 34 35 36 37 38	Maximum Number of Display Stations 39 40	Region Size 41 42	Date/Edit 43	Ideographic Mode 44	Reserved 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	J									

Array Specifications

Sequence Number 1 2 3 4 5	Specification Type 6	Reserved 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	Array Name 27 28 29 30 31 32	Reserved 33 34 35	Number of Entries per Array 36 37 38 39	Length of Entry 40 41 42	Reserved 43	Decimal Positions 44	Sequence (A/D) 45	Reserved 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	E									
	E									
	E									
	E									
	E									

Transaction/Master File Specifications

Sequence Number	Specification Type	File Name	File Definition Location		GET Field Names			Not-Found Indicator	Reserved	Header Record Identifying Indicator	Number of Records	Error Indicator	File Definition Type	Field Level	Reserved
1 2 3 4 5	6	7 8 9 10 11 12 13 14	Library Name 15 16 17 18 19 20 21 22	Source Member Name 23 24 25 26 27 28 29 30	Name 1 31 32 33 34 35 36	Name 2 37 38 39 40 41 42	Name 3 43 44 45 46 47 48	49 50	51 52 53 54	55 56	57 58 59 60	61 62	63	64	65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	T														
	M														
	M														
	M														
	M														
	M														
	M														
	M														
	M														
	M														
	M														

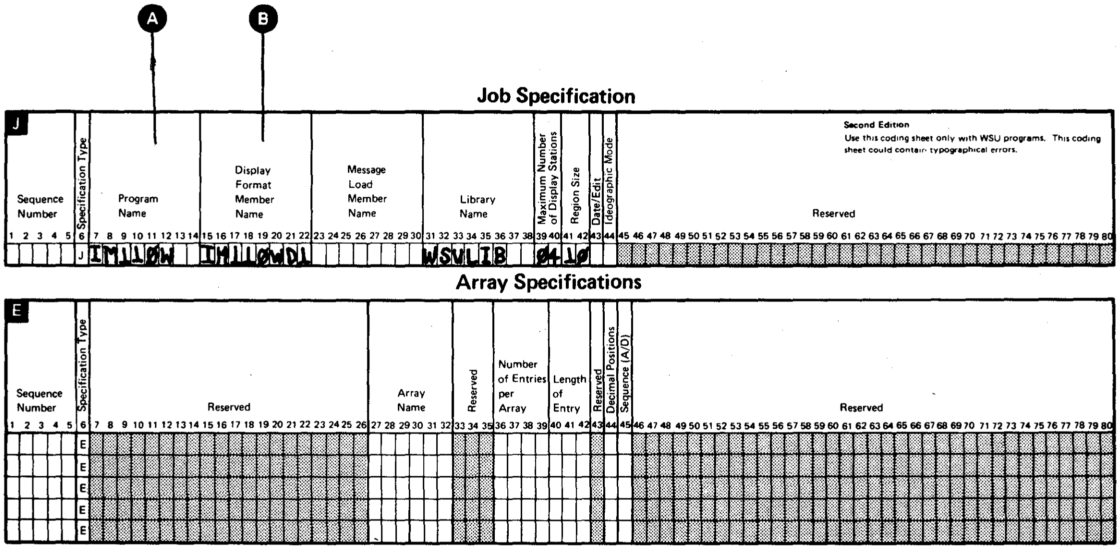
Figure 4-10. J-, E-, T-, and M-Specification Coding Form

J-Specification

All WSU programs must have a J-specification. No matter how complex or simple the program, there is only one J-specification coded.

Two entries are required for the J-specification.

- The *Program Name* (columns 7 through 14) is a name **assigned** to the WSU program *and* to the WSU-generated procedure. Ⓐ
- The load member in which up to 245 display screen **formats are** stored during generation is the *Display Format Member Name* (columns 15 through 22). Ⓑ



Job Specification

Second Edition
Use this coding sheet only with WSU programs. This coding sheet could contain typographical errors.

J Sequence Number 1 2 3 4 5	Specification Type 6	Program Name 7 8 9 10 11 12 13 14	Display Format Member Name 15 16 17 18 19 20 21 22	Message Load Member Name 23 24 25 26 27 28 29 30	Library Name 31 32 33 34 35 36 37 38	Maximum Number of Display Stations 39 40	Region Size 41 42	Date/Edit 43	Ideographic Mode 44	Reserved 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	J	IM110W	IM110WD1		WSULIB	04	10			

Array Specifications

E Sequence Number 1 2 3 4 5	Specification Type 6	Reserved 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	Array Name 27 28 29 30 31 32	Reserved 33 34 35	Number of Entries per Array 36 37 38 39	Length of Entry 40 41 42	Reserved 43	Decimal Positions 44	Sequence (A/D) 45	Reserved 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	E									
	E									
	E									
	E									
	E									

The J-specification and all of its allowed entries are described in Chapter 13, *J-(Job) Specifications.*

T-Specification

The WSU T-specification defines characteristics of the transaction file (a direct file). If the T-specification is not specified in the WSU source program, WSU assumes that the program does not use a transaction file. There can be only one transaction file named for each WSU program. As shown in Figure 4-10, the T-specification is on the same form as the M-specifications because these specifications share common entries.

This section describes entries that you can code on the T-Specification. Chapter 11 contains sample programs for which this specification has been coded. A detailed reference for the entries on the T-specification can be found in Chapter 15, *T-(Transaction File)* and *M-(Master File) Specifications*.

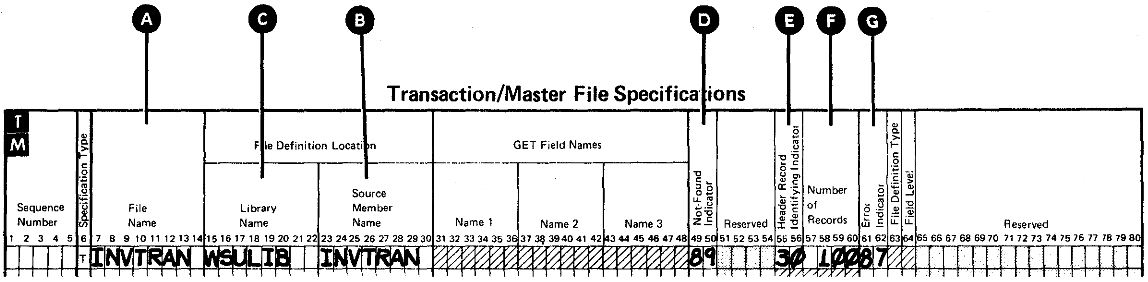
The T-specification requires only two entries: the *File Name* (columns 7 through 14) and the *Source Member Name* (columns 23 through 30). The file name must be the same name that is listed as the transaction file name Ⓐ on the F- and I-specifications. The source member name is the name of the source member that contains the F- and I-specifications for the transaction file. Ⓑ

WSU also checks for the following optional entries on the T-specification:

- *Library Name* (columns 15 through 22) of the source member that contains the F- and I-specifications. Ⓒ
- *Not-Found Indicator* (columns 49 and 50) turns on when WSU cannot determine the record type of the record read from the transaction file or tries to retrieve a record that is beyond the logical end of the transaction file chain for that work station. Ⓓ
- *Header Record Identifying Indicator* (columns 55 and 56) allows the operator to use the Page Backward Group and Page Forward Group command keys, see Chapter 7, *Coding Processing*, and allows the programmer to code GETNH and GETPH operations, see Chapter 18, *Operation Codes*. Ⓔ

- *Number of Records* (columns 57 through 60) to be entered in the transaction file. If this is left blank, storage is allowed for 1000 records. Ⓕ

- *Error Indicator* (columns 61 and 62) turns on when an output/input error occurs for a transaction file operation or a record-not-found condition occurs for a transaction file and an indicator has not been coded in columns 49 and 50 of the T-specification. If there is an error, the code for this error is placed in *ERROR, a WSU reserved field. See Chapter 3, *Reserved Fields and Indicators*, for a discussion of *ERROR. Ⓖ



Transaction/Master File Specifications

Sequence Number 1 2 3 4 5	Specification Type 6	File Name 7–14	File Definition Location: Library Name 15–22	File Definition Location: Source Member Name 23–30	GET Field Names: Name 1 31–36	GET Field Names: Name 2 37–42	GET Field Names: Name 3 43–48	Not-Found Indicator 49 50	Reserved 51–54	Header Record Identifying Indicator 55 56	Number of Records 57–60	Error Indicator 61 62	File Definition Type 63	Field Level 64	Reserved 65–80
	T	INVTRAN	WSULIB	INVTRAN				89		30	100	87			

M-Specification

An M-specification provides information about a master file, session-level fields, or local-data-area fields. As many as 20 master files, one group of session-level fields, and one group of local-data-area fields can be used in a WSU program.

As shown in Figure 4-10, M-specifications are on the same form as the T-specification.

Each M-specification provides:

- *File Name* (columns 7 through 14) names the master file, a group of session-level fields, or a local data area. The master file is the same file which appears in columns 7 through 14 of the F- and I-specifications. The name of session-level fields or the name of a local data area is shown also on the I-specification. Ⓐ

  If the local data area is defined in the WSU program on the I-specification and the M-specification, WSU automatically reads a display station's local data area when the display station signs on and updates this data area when the display station signs off.

  Session-level fields are defined on the WSU M-specification as well as on the I-specification. These session-level fields are initialized when a work session is started at the display station.

- *Library Name* (columns 15 through 22) names the library that contains the source member. Ⓑ

- *Source Member Name* (columns 23 through 30) names the source member that contains the F- and I-specifications for the master file, the I-specifications for the group of session-level fields or for the local data area. Ⓒ

- *GET Field Names* (columns 31 through 48) names the input fields that WSU combines (in the order you code them) to search the key field in the master file. This field is used to access records (by means of GET and PUT operations) in a master file. Ⓓ

  Example:

  I-specification defines: FIELDA and FIELDB

  C-specification defines (in a result field): FIELDC

  M-specification defines: FIELDA FIELDB FIELDC

  When a GET is issued, assume the contents of these fields are:

  FIELDC: A124

  FIELDB: 123

  FIELDA: ABC

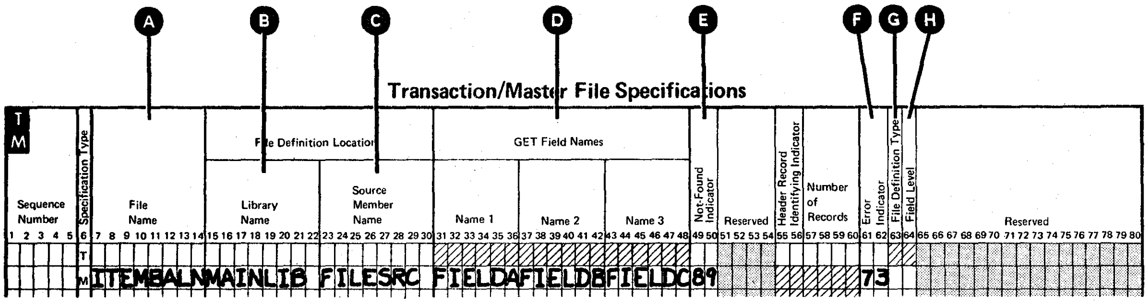
  The field that is formed to access the record is: A124123ABC

- *Not-Found Indicator* (columns 49 and 50) turns on when WSU cannot determine the type of the record read from the master file or when there is no record that has the specified key. Ⓔ

- *Error Indicator* (columns 61 and 62) turns on when an output/input error occurs during a file operation for a master file or when a record-not-found condition occurs for the master file and an indicator has not been coded in *Not-Found Indicator* of the M-specification. If there is an error, the code for the error is placed in *ERROR, a WSU reserved field. Ⓕ

- *File Definition Type* (column 63) indicates use of a local data area, group of fields, or master file. Ⓖ

- *Field Level* (column 64) indicates session-level local-data-area fields, a group of session-level fields, mode-level local data-area fields, or master file fields that are mode level. Ⓗ



Transaction/Master File Specifications

Sequence Number 1 2 3 4 5	Specification Type 6	File Name 7–14	File Definition Location: Library Name 15–22	File Definition Location: Source Member Name 23–30	GET Field Names: Name 1 31–36	GET Field Names: Name 2 37–42	GET Field Names: Name 3 43–48	Not-Found Indicator 49 50	Reserved 51–54	Header Record Identifying Indicator 55 56	Number of Records 57–60	Error Indicator 61 62	File Definition Type 63	Field Level 64	Reserved 65–80
	T														
	M	ITEMBALN	MAINLIB	FILESRC	FIELDA	FIELDB	FIELDC	89				73			

Chapter 5. Coding Arrays

An array is a continuous series of data fields stored side by side so they can be referenced as a group or individually. In an array, each individual data field is called an *element*. Figure 5-1 shows an array of 12 elements containing the total sales for each month of the year.

Each element of the array has the same characteristics; that is, each contains data:

- In the same format (alphameric or numeric)
- Of the same length
- With the same number of decimal positions.

Within a numeric array, elements may be positive or negative.

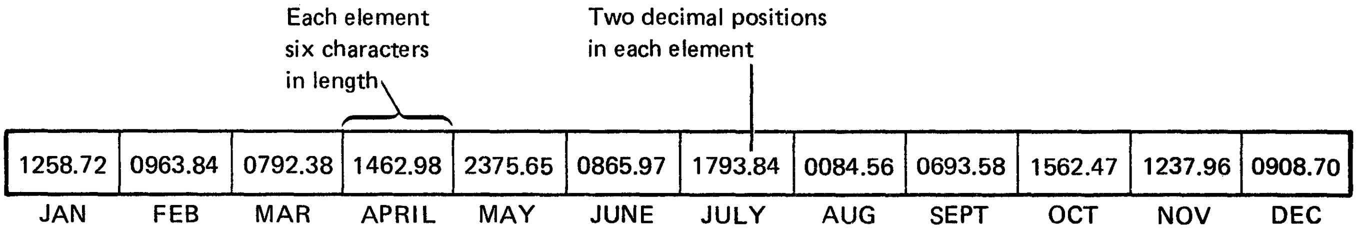


Figure 5-1. A Numeric Array

Using Arrays

Arrays are generally used for storing variable transaction data, and for storing totals from the variable data that are used independently.

You can use arrays effectively when you want to:

- Reference all elements at one time.

  Arrays can reduce the number of specifications you must code for such a program (see Figure 5-2) as well as the time required to reference the entries.

- Directly reference a data item within a group of items.

  Individual array elements can be referenced quickly by specifying an *index* (or subscript) that points to the item.

- Handle data within a field.
- Search a group of items for one or more that meet a certain requirement.
- Total the values of a group of data items.
- Sort a group of data items into ascending or descending order.

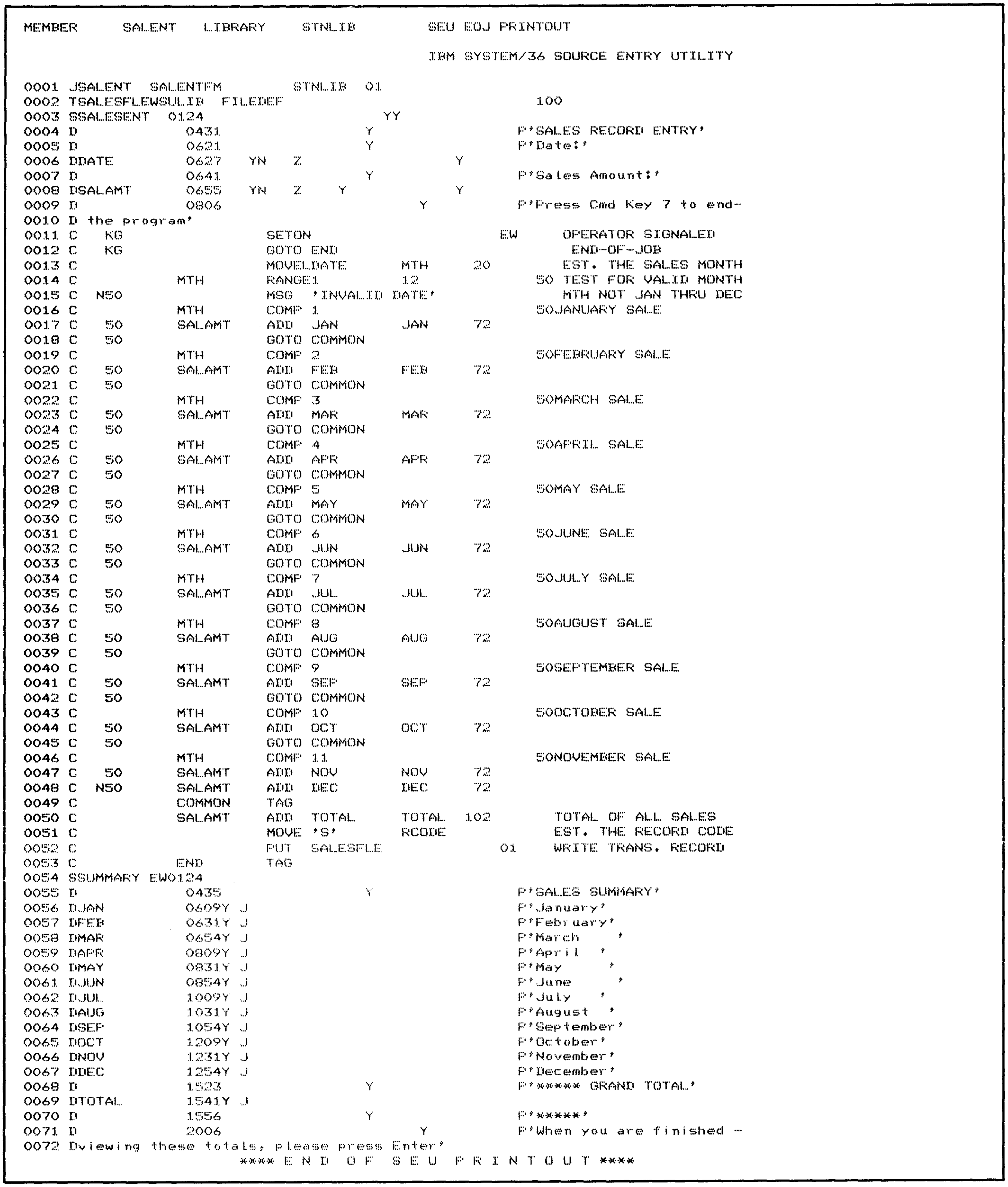
The next two figures show how the number of specifications can be reduced for a program when you use array processing. These figures are SEU printouts, so there are no F- or I-specifications shown.

Suppose your company is in the process of converting from a manual operation to one that uses a System/36, and you need to enter several hundred sales records from the prior year into the system. While the records need not be in any sequence, management is interested in the sales totals for each month, as well as the total sales for the year.

If a new programmer who did not know how to use WSU array processing were assigned the task of writing the data entry program, the results might be shown as in Figure 5-2.

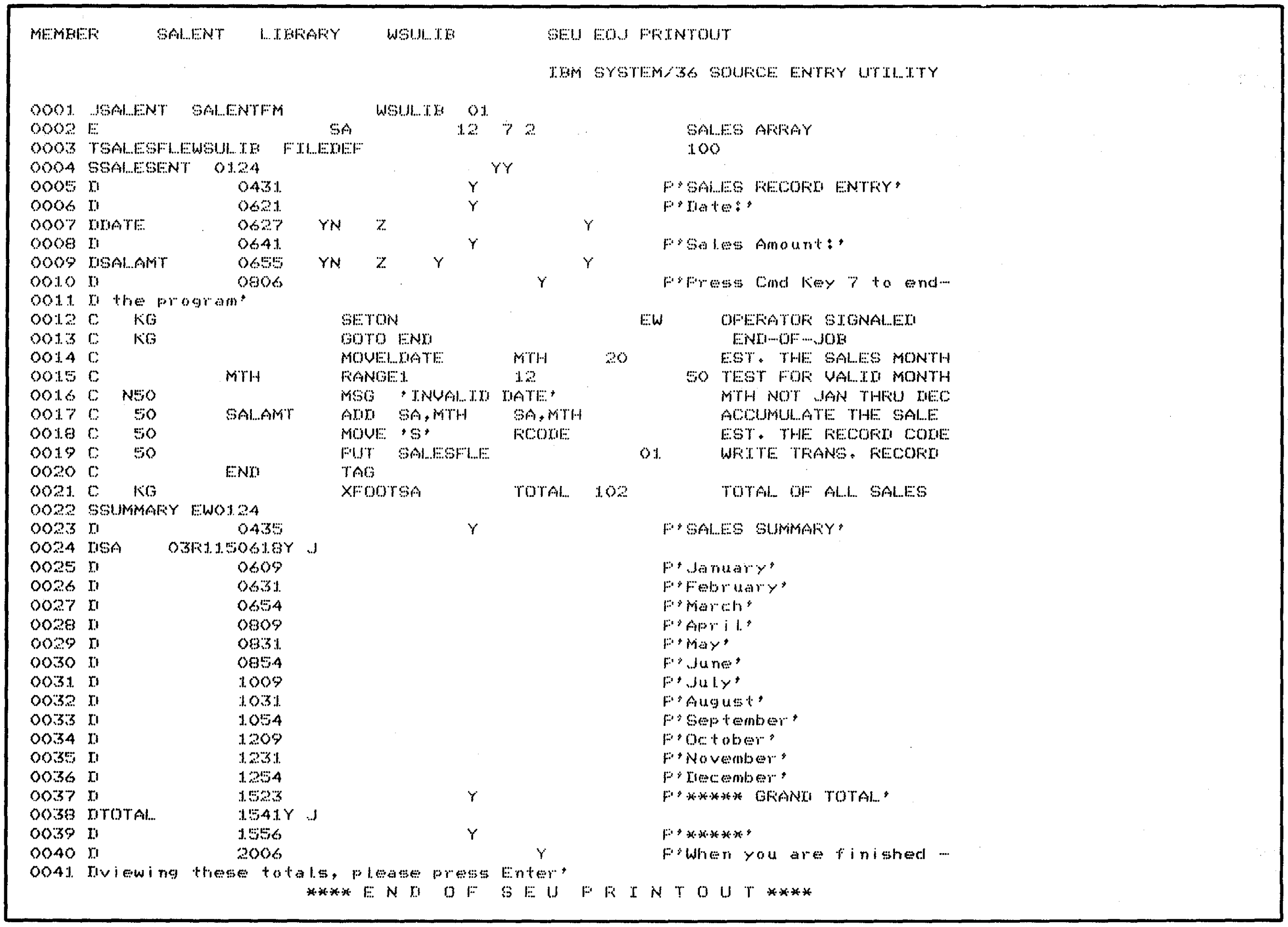
However, if the program were written using WSU array processing, a significant number of specification statements could be saved, as shown in Figure 5-3.

Compare the two listings and notice that the first five statements are unchanged but that statements 16 through 49 (34 instructions) are replaced by *one instruction*!



```
MEMBER     SALENT   LIBRARY    STNLIB         SEU EOJ PRINTOUT

                                             IBM SYSTEM/36 SOURCE ENTRY UTILITY

JSALENT  SALENTFM        STNLIB  01
TSALESFLEWSULIB  FILEDEF                            100
SSALESENT  0124                    YY
D            0431                Y                F'SALES RECORD ENTRY'
D            0621                Y                F'Date:'
DDATE        0627  YN   Z                 Y
D            0641                Y                F'Sales Amount:'
DSALAMT      0655  YN   Z    Y            Y
D            0806                     Y           F'Press Cmd Key 7 to end-
D the program'
C   KG                SETON                     EW     OPERATOR SIGNALED
C   KG                GOTO END                          END-OF-JOB
C                     MOVELDATE      MTH     20        EST. THE SALES MONTH
C           MTH       RANGE1         12             50 TEST FOR VALID MONTH
C  N50                MSG 'INVALID DATE'               MTH NOT JAN THRU DEC
C           MTH       COMP 1                        50JANUARY SALE
C   50      SALAMT    ADD  JAN       JAN     72
C   50                GOTO COMMON
C           MTH       COMP 2                        50FEBRUARY SALE
C   50      SALAMT    ADD  FEB       FEB     72
C   50                GOTO COMMON
C           MTH       COMP 3                        50MARCH SALE
C   50      SALAMT    ADD  MAR       MAR     72
C   50                GOTO COMMON
C           MTH       COMP 4                        50APRIL SALE
C   50      SALAMT    ADD  APR       APR     72
C   50                GOTO COMMON
C           MTH       COMP 5                        50MAY SALE
C   50      SALAMT    ADD  MAY       MAY     72
C   50                GOTO COMMON
C           MTH       COMP 6                        50JUNE SALE
C   50      SALAMT    ADD  JUN       JUN     72
C   50                GOTO COMMON
C           MTH       COMP 7                        50JULY SALE
C   50      SALAMT    ADD  JUL       JUL     72
C   50                GOTO COMMON
C           MTH       COMP 8                        50AUGUST SALE
C   50      SALAMT    ADD  AUG       AUG     72
C   50                GOTO COMMON
C           MTH       COMP 9                        50SEPTEMBER SALE
C   50      SALAMT    ADD  SEP       SEP     72
C   50                GOTO COMMON
C           MTH       COMP 10                       50OCTOBER SALE
C   50      SALAMT    ADD  OCT       OCT     72
C   50                GOTO COMMON
C           MTH       COMP 11                       50NOVEMBER SALE
C   50      SALAMT    ADD  NOV       NOV     72
C  N50      SALAMT    ADD  DEC       DEC     72
C           COMMON    TAG
C           SALAMT    ADD  TOTAL     TOTAL  102       TOTAL OF ALL SALES
C                     MOVE 'S'       RCODE            EST. THE RECORD CODE
C                     PUT  SALESFLE            01     WRITE TRANS. RECORD
C           END       TAG
SSUMMARY EW0124
D            0435                Y                F'SALES SUMMARY'
DJAN         0609Y J                              F'January'
DFEB         0631Y J                              F'February'
DMAR         0654Y J                              F'March    '
DAPR         0809Y J                              F'April  '
DMAY         0831Y J                              F'May     '
DJUN         0854Y J                              F'June     '
DJUL         1009Y J                              F'July   '
DAUG         1031Y J                              F'August  '
DSEP         1054Y J                              F'September'
DOCT         1209Y J                              F'October'
DNOV         1231Y J                              F'November'
DDEC         1254Y J                              F'December'
D            1523                Y                F'***** GRAND TOTAL'
DTOTAL       1541Y J
D            1556                Y                F'*****'
D            2006                     Y           F'When you are finished -
Dviewing these totals, please press Enter'
                   **** E N D  O F  S E U  P R I N T O U T ****
```

Figure 5-2. Data Entry Program WITHOUT Array Processing



```
MEMBER     SALENT   LIBRARY    WSULIB        SEU EOJ PRINTOUT

                                             IBM SYSTEM/36 SOURCE ENTRY UTILITY

0001 JSALENT  SALENTFM        WSULIB  01
0002 E                    SA         12  7 2             SALES ARRAY
0003 TSALESFLEWSULIB  FILEDEF                            100
0004 SSALESENT  0124                    YY
0005 D            0431                Y                P'SALES RECORD ENTRY'
0006 D            0621                Y                P'Date:'
0007 DDATE        0627   YN   Z                 Y
0008 D            0641                Y                P'Sales Amount:'
0009 DSALAMT      0655   YN   Z    Y            Y
0010 D            0806                      Y          P'Press Cmd Key 7 to end-
0011 D the program'
0012 C   KG                SETON                     EW     OPERATOR SIGNALED
0013 C   KG                GOTO END                          END-OF-JOB
0014 C                     MOVELDATE      MTH     20        EST. THE SALES MONTH
0015 C          MTH        RANGE1         12             50 TEST FOR VALID MONTH
0016 C  N50                MSG  'INVALID DATE'              MTH NOT JAN THRU DEC
0017 C   50     SALAMT     ADD  SA,MTH    SA,MTH            ACCUMULATE THE SALE
0018 C   50                MOVE 'S'       RCODE             EST. THE RECORD CODE
0019 C   50                PUT  SALESFLE             01     WRITE TRANS. RECORD
0020 C          END        TAG
0021 C   KG                XFOOTSA        TOTAL  102        TOTAL OF ALL SALES
0022 SSUMMARY EW0124
0023 D            0435                Y                P'SALES SUMMARY'
0024 DSA    03R1150618Y J
0025 D            0609                                 P'January'
0026 D            0631                                 P'February'
0027 D            0654                                 P'March'
0028 D            0809                                 P'April'
0029 D            0831                                 P'May'
0030 D            0854                                 P'June'
0031 D            1009                                 P'July'
0032 D            1031                                 P'August'
0033 D            1054                                 P'September'
0034 D            1209                                 P'October'
0035 D            1231                                 P'November'
0036 D            1254                                 P'December'
0037 D            1523                Y                P'***** GRAND TOTAL'
0038 DTOTAL       1541Y J
0039 D            1556                Y                P'*****'
0040 D            2006                      Y          P'When you are finished -
0041 Dviewing these totals, please press Enter'
                        **** E N D   O F   S E U   P R I N T O U T ****
```

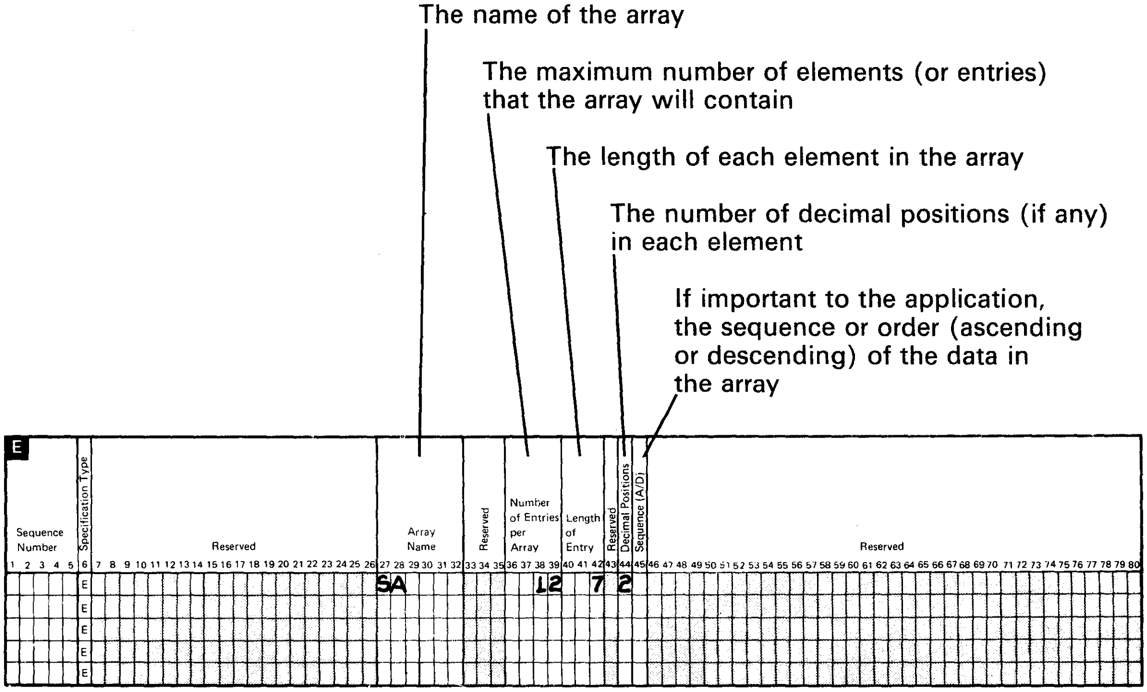
Figure 5-3. Data Entry Program WITH Array Processing

Defining Arrays

WSU uses the kind of arrays that RPG calls *execution-time arrays*; that is, the arrays are built (or loaded) when the program is run. The data for such arrays may be read from input file records at the beginning of the program, generated by processing operations during the course of the program's run, or filled by data entered by the operator.

A WSU array can be mode-level, session-level, or job-level.

You tell the WSU program that you wish to set up an array by coding *E-(array) specifications* that define:



The name of the array

The maximum number of elements (or entries) that the array will contain

The length of each element in the array

The number of decimal positions (if any) in each element

If important to the application, the sequence or order (ascending or descending) of the data in the array

(The above E-specification defines the array pictured in Figure 5-1.)

E-specifications *reserve* the appropriate space in storage for the array. The array is then *loaded* from data produced by operations described on C-(processing) specifications, from input data described on I-(input) specifications (see *Loading Arrays* later in this chapter), or from data entered by the operator.

Naming Arrays

The name used to reference an array can be up to 6 characters long. The name of a job-level array must begin with the & character; all other array names must begin with an alphabetic character.

Keep in mind, however, that if you plan to use an index to reference individual elements of an array (in some place other than factor 1 or factor 2 of the C-specification), the *name* also includes the index and the comma separating it from the array name. Therefore, try to keep the array name short, so that the array name, the comma, and the index do not exceed 6 characters. See *Indexing an Array* later in this chapter.

Spacing Arrays

A WSU array can contain as many as 9999 elements. *Alphameric* elements can be up to 256 characters long; *numeric* elements can be as long as 15 digits, with a maximum of 9 decimal positions.

However, because WSU arrays are part of the program that is running, using a very large array increases the time it takes to run the program as well as the amount of storage required to run the program.

Also, if you plan to display all or part of the array, you should keep in mind how the elements will fit in rows or columns on the display (see *Displaying Arrays* later in this chapter).

Ordering Arrays

WSU arrays are not sequence-checked and therefore need not be in any sequential order. However, if you plan to use a LOKUP operation to search for high or low conditions, you must specify whether the array is (or should be) in ascending or descending order.

If at the time you want to do a high or low lookup you are not sure that the elements in the array are properly ordered, you can use the SORTA operation to put the elements in the sequence you define on the E-specification. If you have ideographic characters in the array, the SORTA cannot properly sequence the elements.

Referencing Entire Arrays For Processing

You reference entire arrays on C-specifications by using the array name as factor 1, factor 2, or the result field; the name actually refers to all of the array elements. An operation specified for an entire array is done *to each element* of the array, using the same field, constant, or literal.

You can use the MOVE and MOVEL operations and all of the arithmetic operations (ADD, SUB, MULT, DIV, Z-ADD, Z-SUB) *except MVR* to process entire arrays. In addition, WSU provides four special *for-arrays-only* operations:

- MOVEA, which moves data to or from an array
- XFOOT, which sums the values of array elements
- SORTA, which sorts array elements
- LOKUP, which searches for a specific array element.

An operation done on all elements yields more than one result; therefore, except with the XFOOT and LOKUP operations, you don't use resulting indicators when you process entire arrays.

Processing from Array-to-Array

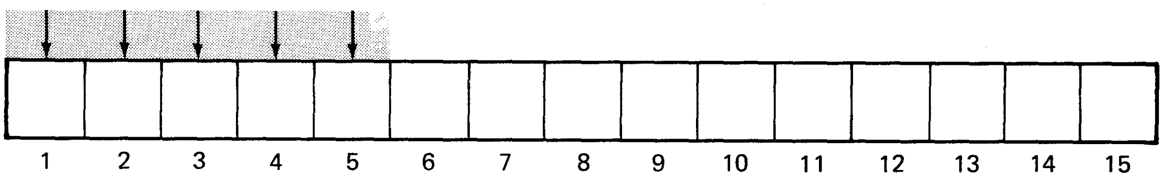
Only one C-specification is necessary to do most array-to-array processing, because, as stated before, the name of an array actually refers to all of the elements in that array. Refer to Figure 5-4 for the E-specifications of the DAY array.

Suppose you build an array named DAY from a file named SALES, whose records contain (among other things) the day's sales totals for each employee:

Clerk 1 Sales	Clerk 2 Sales	Clerk 3 Sales	Clerk 4 Sales	Clerk 5 Sales	Clerk 6 Sales	Clerk 7 Sales	Clerk 8 Sales

Clerk 9 Sales	Clerk 10 Sales	Clerk 11 Sales	Clerk 12 Sales	Clerk 13 Sales	Clerk 14 Sales	Clerk 15 Sales	

DAY Array:



To accumulate weekly totals into another array, named WEEK, you just code a processing operation that adds the DAY array to the WEEK array and places the result back in the WEEK array:

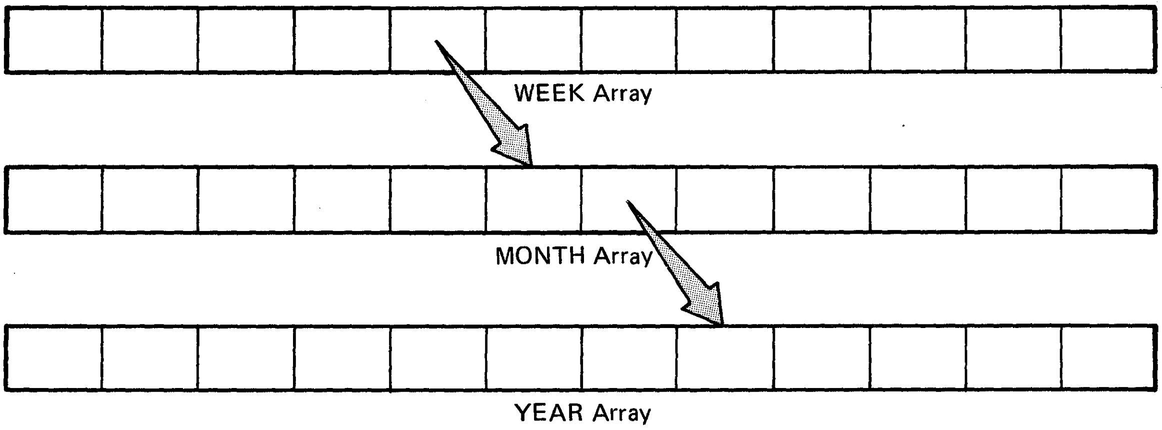
Sequence Number	Specification Type	Processing Function	Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Pos	Half Adjust (H)	Resulting Indicators	Comments
	C			WEEK	ADD	DAY	WEEK					ACCUM SALES

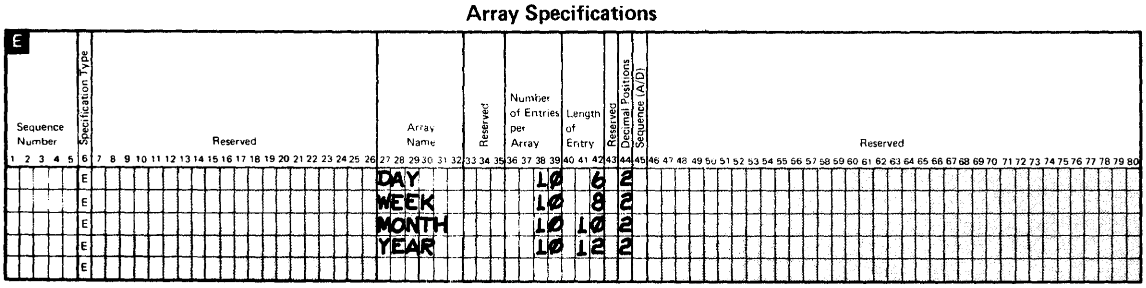
Refer to Figure 5-4 for the E-specifications for the WEEK array.

What happens is that WSU adds the first element of DAY to the first element of WEEK and places the result back in the first element of WEEK, then adds the second elements, and so on:

	1	2	3	4	5	6	7	8	9	10
DAY	0015.21	0012.86	0025.31	0008.93	0017.83	0019.24	0015.67	0032.81	0042.21	0021.87
	+	+	+	+	+	+	+	+	+	+
WEEK	0072.18	0142.96	0063.90	0089.61	0076.95	0128.76	0134.21	0062.34	0079.83	0052.24
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
WEEK	0087.39	0155.82	0089.21	0098.54	0094.78	0148.00	0149.88	0095.15	0122.04	0074.11

To carry this idea further, one of the most common uses of arrays is to accumulate more than one group of totals. Such a procedure is called *rolling totals,* because one total is used to obtain a greater total, which is then used to calculate an even larger total, and so on. Each total is *rolled into,* or accumulated, into the next total. To continue the previous example, you could add the WEEK array to an array called MONTH, and then add the MONTH array to one called YEAR, and so forth. Refer to Figure 5-4 for the E-specifications for the MONTH and YEAR arrays.





Array Specifications

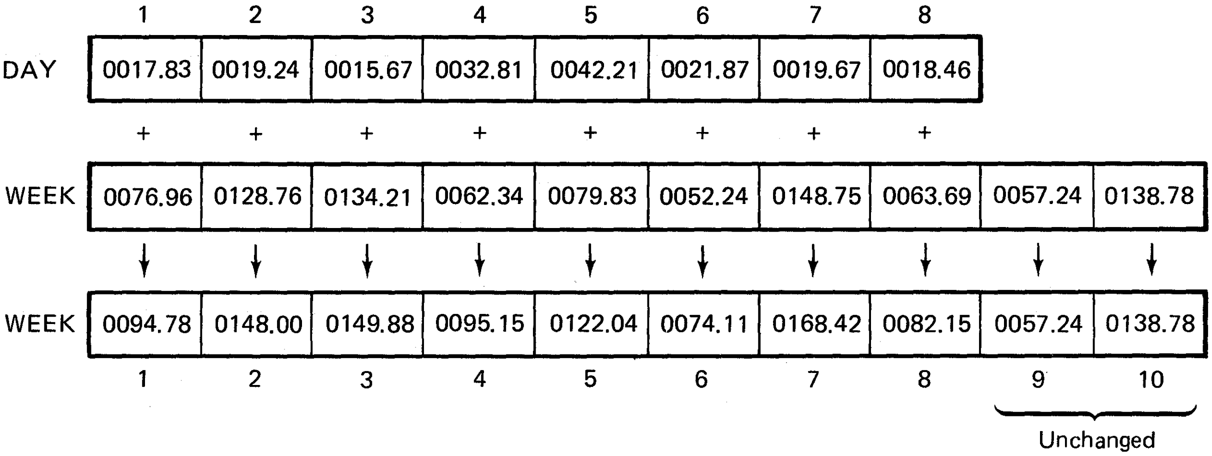
E Sequence Number 1 2 3 4 5	Specification Type 6	Reserved 7–26	Array Name 27–32	Reserved 33 34 35	Number of Entries per Array 36 37 38 39	Length of Entry 40 41 42	Reserved 43	Decimal Positions 44	Sequence (A/D) 45	Reserved 46–80
	E		DAY		10	6		2		
	E		WEEK		10	8		2		
	E		MONTH		10	10		2		
	E		YEAR		10	12		2		
	E									

Figure 5-4. E-Specifications for DAY, WEEK, MONTH, and YEAR

Processing Arrays of Different Lengths

In the earlier example, all arrays used in an operation were of the same length: factor 1 (DAY), factor 2 (WEEK), and the result array (WEEK) each contained 10 elements. Thus, the operations were carried out until all elements were processed.

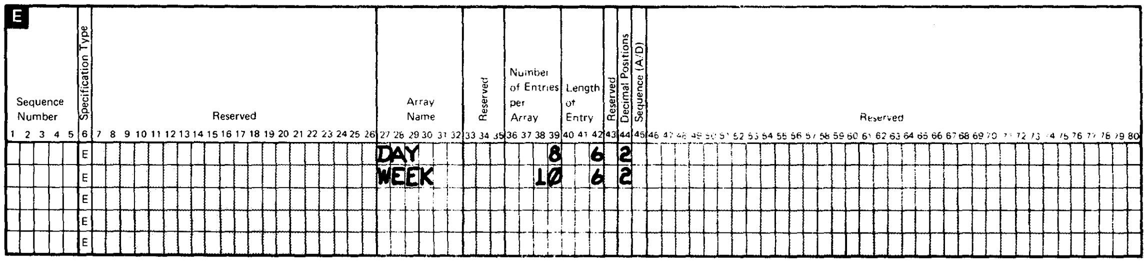
Suppose, though, that the DAY array contains only eight elements, while the WEEK array contains 10. In such a case, the operations are done only until the last element in the shortest array has been processed. Thus, the eight elements of DAY are added to the first eight elements of WEEK, and the eight results are placed in the first eight elements of WEEK. The remaining two elements of the result field (WEEK) remain unchanged.



Likewise, if the result array is shorter than any of the factors (arrays), the operation is repeated only for the number of elements in the shortest (result) array.

The E-specifications for the arrays, DAY and WEEK, are:

Array Specifications



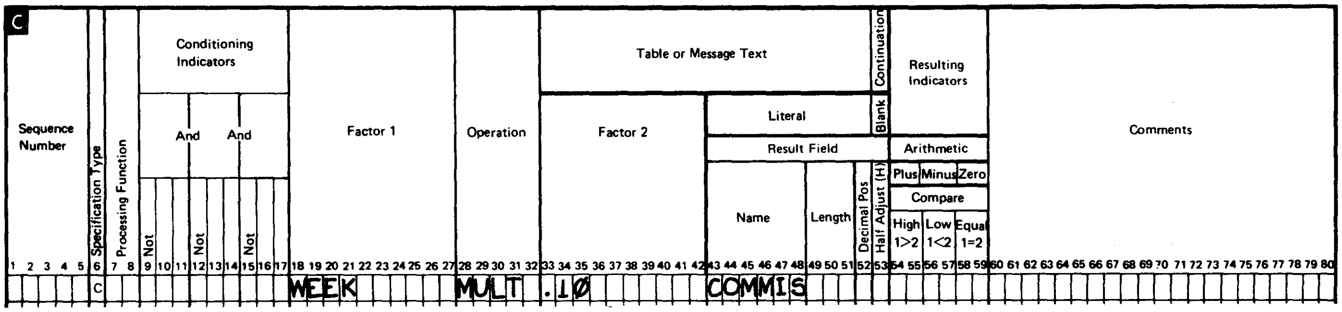
Sequence Number (1-5)	Specification Type (6)	Reserved (7-26)	Array Name (27-32)	Reserved (33-35)	Number of Entries per Array (36-39)	Length of Entry (40-42)	Reserved (43)	Decimal Positions (44)	Sequence (A/D) (45)	Reserved (46-80)
	E		DAY		8	6		2		
	E		WEEK		10	6		2		
	E									
	E									
	E									

Processing an Array with a Single Value

Another way in which you can process an entire array is by adding the same value to every element in the array (or subtracting, multiplying, dividing using the same value).

Suppose the employees in the earlier example are to receive a commission of 10% of their sales, to be paid at the end of the week. After all daily sales have been accumulated into the WEEK array, you want to multiply each of the 10 elements in WEEK by the value .10 and to place the commission amounts in another 10-element array called COMMIS.

When one of the factors in an array operation is a constant or a field containing a value, the operation is done using the same constant or field on every element in the array. Thus, you would specify:



```
     C           WEEK      MULT .10       COMMIS
```

WEEK Array

873.90	155.82	892.10	985.40	947.80	148.00	149.88	951.50	122.04	741.11

X.10

COMMIS Array

87.39	15.58	89.21	98.54	94.78	14.80	14.99	95.15	12.20	74.11

If you were doing the same example but using one array, you would specify:

C

Sequence Number	Specification Type	Processing Function	Conditioning Indicators		And		And		Factor 1	Operation	Table or Message Text						Continuation	Resulting Indicators			Comments
											Factor 2	Literal					Blank	Arithmetic			
												Result Field						Plus	Minus	Zero	
												Name	Length	Decimal Pos	Half Adjust (H)			Compare			
			Not		Not		Not											High 1>2	Low 1<2	Equal 1=2	
1 2 3 4 5	6	7 8	9	10 11	12	13 14	15	16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53			54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	C								WEEK	MULT	1.10	WEEK									

WEEK Array

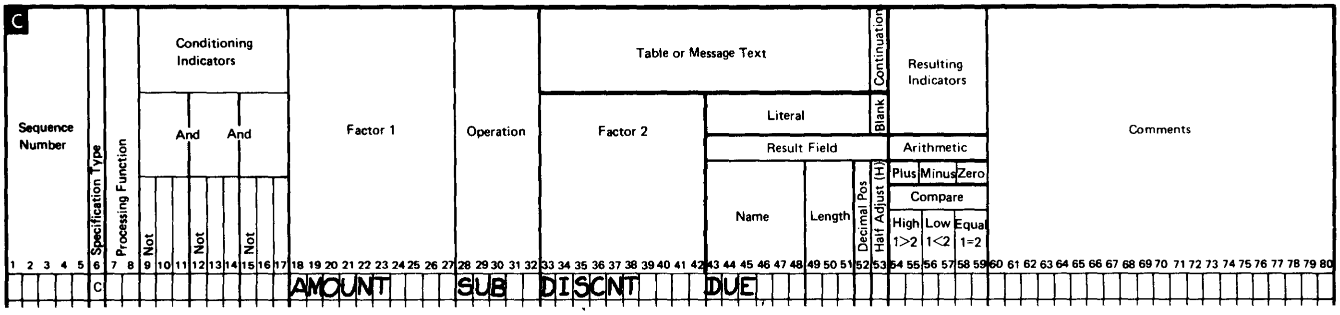
873.90	155.82	822.10	985.40	947.80	148.00	149.88	951.50	122.04	741.11

X1.10

WEEK Array

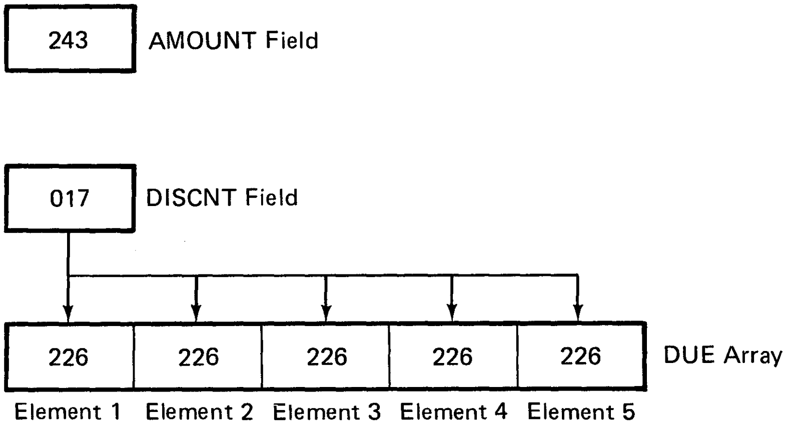
961.29	171.40	904.31	1083.94	1042.58	162.80	164.87	1046.65	134.24	815.22

In another example of using a single value for processing an array, a company is giving a 17% discount on all merchandise. You can also use a field or constant as *both* factors to place the same result in every element of an array in this example. The following C-specification shows the single field named DISCNT being subtracted from the single field AMOUNT, with the result placed in a five-element array named DUE.



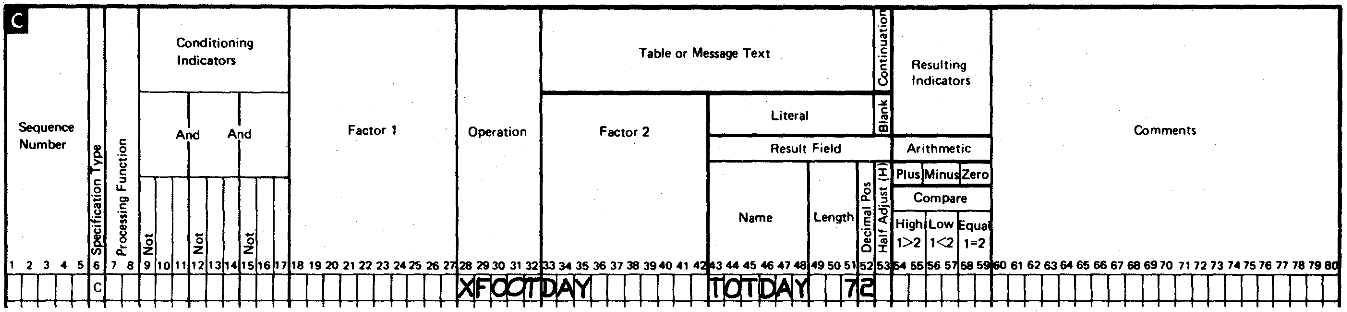
```
     C           AMOUNT    SUB  DISCNT    DUE
```

The value (017) in DISCNT is subtracted from the value (243) in AMOUNT, and the result (226) is placed in each of the five elements of the DUE array.

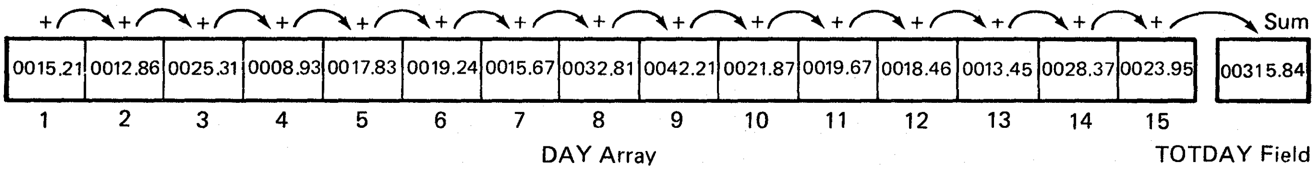


Adding All Elements within an Array

Suppose, in addition to accumulating a monthly sales total for each employee, you also want to know the total of all sales each day. To obtain that total, you can use the XFOOT operation to add together (or sum) the values of all elements in the DAY array and then place the total in a single field called TOTDAY:



```
     C                     XFOOTDAY       TOTDAY  72
```



0015.21	0012.86	0025.31	0008.93	0017.83	0019.24	0015.67	0032.81	0042.21	0021.87	0019.67	0018.46	0013.45	0028.37	0023.95	00315.84
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

DAY Array

TOTDAY Field

In most types of array operations, as many results are produced as there are elements in the array. However an XFOOT operation produces only one result, the total of all elements; so you specify a *single field name* rather than array name for the result field.

Also, because there is only one result, you can assign a resulting indicator to determine if the total is plus, minus, or zero. In this example, no resulting indicator was specified because the sales amounts will always be positive.

Referencing Individual Array Elements in Processing

In addition to referencing all elements of an array, you can use an individual array element in processing. You can use all of the same operations on elements that you use on entire arrays, except for XFOOT and SORTA. In addition, you can also use the MVR and COMP operations as well as resulting indicators, because each operation *on an element* produces only one result.

Indexing an Array

As mentioned earlier, if a processing operation specifies an array name alone, the operation is automatically performed for *every* element of the array. To reference only a *single* element of an array, you must identify that element by placing a comma after the array name, followed by an *index* that points to the particular element. This index can be either:

- The *actual number* of the element to be referenced (for example: ARY,9)
- The name of a *field containing the number* of the element to be used (for example: ARY,IX).

You will recall from *Defining Arrays* earlier in this chapter that the name used to refer to an array cannot be longer than 6 characters; and that when referencing individual fields, both the array name and an index are necessary to refer to the data. Therefore, the array name *plus* the comma *plus* the index cannot exceed 6 characters, *unless* the reference is specified *only* as factor 1 or 2. For factors 1 and 2, the total reference can be up to 10 characters long; however, the *array name* portion of the reference still can be no more than 6 characters (for example: SALES,INDX).

Note: When you are referencing an array element, you *must* make sure that the index you use is an allowed one. If you use a 0, a number greater than the length of your array, or a negative number, your program will have an abnormal ending.

Specifying a Constant Index

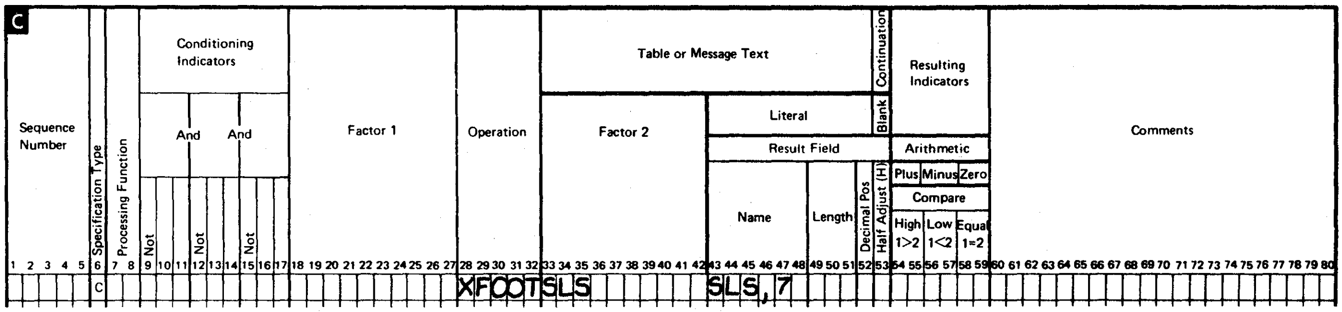
If you know exactly which element is to be used in an operation and if the same element is to be referenced every time, you can use a *constant* as the index.

Assume you have defined a seven-element array named SLS to contain an employee's six daily commission amounts and the total commission for the week. The array elements are 5-digit numbers with two decimal positions.

The six daily amounts from one employee's input records are read into elements 1 through 6 of the array. The seventh field on the input record contains zeros and is read into element 7 of the array.

03520	02780	04237	03187	01790	02593	00000	SLS Array
1	2	3	4	5	6	7	

Once the data is in the array, you would use the XFOOT operation to add all elements of the array and place the total in the seventh element:

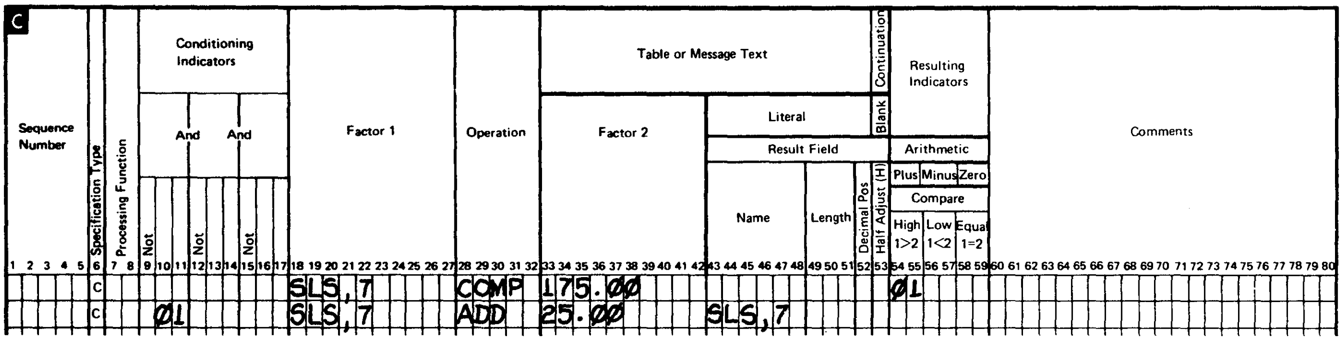


```
     C                     XFOOTSLS       SLS,7
```

03520	02780	04237	03187	01790	02593	18107	SLS Array
						Element 7	

The weekly total for every sales representative is always stored in the seventh element; therefore, the actual number 7 can be specified as the index.

Also, a $25 bonus is to be added to a representative's total if the weekly commission exceeds $175. Thus, for every record processed, element 7 must first be compared to $175 to determine if the bonus is to be added to the contents of element 7:



Specification Type (6)	Conditioning Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Resulting Indicators High 1>2 (54-55)
C		SLS,7	COMP	175.00		01
C	01	SLS,7	ADD	25.00	SLS,7	

SLS Array	03520	02780	04237	03187	01790	02593	18107

+ 2500

SLS Array	03520	02780	04237	03187	01790	02593	20607

Element 7

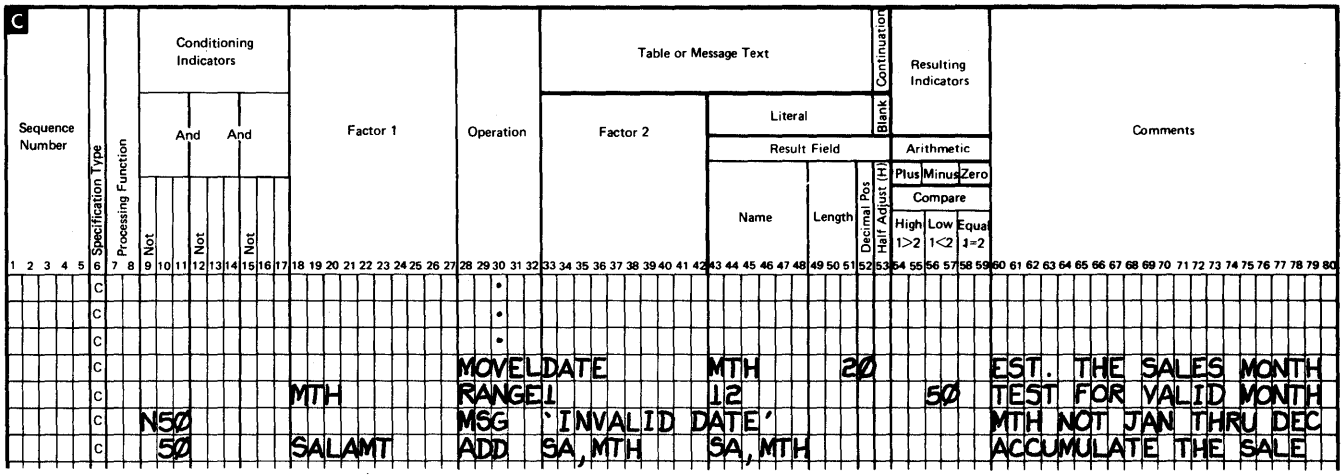
Specifying a Variable Index

On the other hand, if the array element will be different whenever a particular operation is performed, the index should be a field name rather than an actual number. In this way, the number stored in the index field can be changed during the program to indicate which array element is to be referenced.

To repeat the example situation used earlier in Figure 5-2: Suppose again that your company is in the process of converting from a manual operation to one that uses a System/36, and you need to enter several hundred sales records from the prior year into the system. While the records need not be in any sequence, management is interested in the sales totals for each month, as well as the total sales for the year.

You might use an array called SA to contain the monthly sales total: element 1 contains the sales for January, element 2 for February, and so on. When a sales record is entered, the total for that month must be updated to reflect the current total. This means you should reference (add to) only particular elements of the array.

The contents of the SALAMT field (as shown in the C-specification below) must be added to one of the SA array elements for every record entered; therefore, an index must be used to reference only the individual element to be updated. Each daily record is for a different month, so the array element to be increased will vary each time the operation is done. An actual number cannot be specified as the index, because the SALAMT value would be added to the same element for every sale. Instead, you can move the leftmost two digits of the DATE field (the ones containing the month number, as in 100882) into a 2-digit index field named MTH. Then, every time the ADD operation is done, the part of the date just stored in MTH indicates which element of the array is to be referenced.

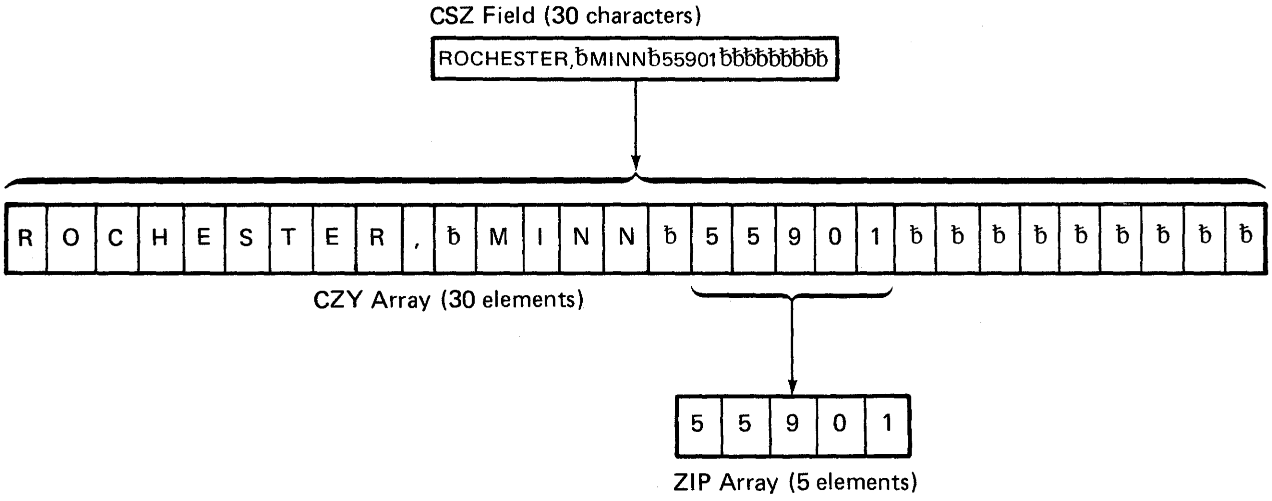


Sequence Number	Specification Type	Processing Function	Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Pos	Half Adjust (H)	High 1>2	Low 1<2	Equal 1=2	Comments
	C				.									
	C				.									
	C				.									
	C				MOVEL	DATE	MTH	2	0					EST. THE SALES MONTH
	C			MTH	RANGE	1	12					50		TEST FOR VALID MONTH
	C		N50		MSG	'INVALID DATE'								MTH NOT JAN THRU DEC
	C		50	SALAMT	ADD	SA,MTH	SA,MTH							ACCUMULATE THE SALE

Referencing Only Part of a Field

When a field is referenced in an operation, all characters within that field are used in the processing. However, you may wish to reference only some of the data stored in a field. For example, an address field contains the city, state, and zip code, but you want to reference only the zip code.

You can use the indexing capability of arrays to reference specific characters from an input field. You do this by setting up two arrays: one to contain the entire field of data, and one to hold only the specific characters you want to reference. First, the entire field from which you wish to use data is stored in another array, which was previously defined as containing as many 1-byte elements as there are characters in the field to be referenced:



Thus, each character of the one field is actually stored in a separate element of the array, and the array elements can then be referenced one at a time, using an index, until an element containing a specific character is located. This process of checking the elements of an array for particular data is called *field scanning*.

After scanning the elements and locating a specific character, you can then move that character and/or any characters (elements) on either side of it to a smaller array. This array will then contain the portion of the original input field that you wish to reference separately in processing operations.

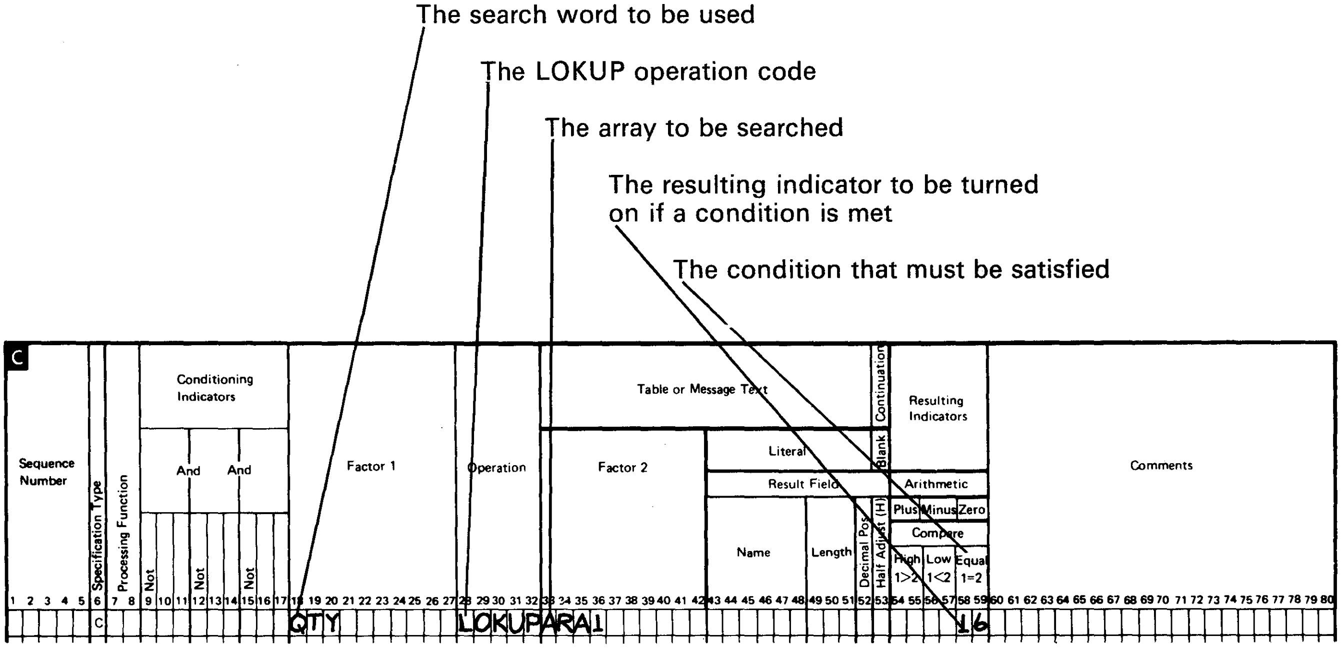
The two arrays for this program are defined with the E-specifications in the following example. CZY is set up to contain 30 elements, one for each character of the CSZ field from the input record. The five elements of the ZIP array will be used to contain the zip code portion of the CSZ field.

Array Specifications

Sequence Number 1 2 3 4 5	Specification Type 6	Reserved 7–26	Array Name 27–32	Reserved 33–35	Number of Entries per Array 36–39	Length of Entry 40–42	Reserved 43	Decimal Positions 44	Sequence (A/D) 45	Reserved 46–80
	E		CZY		30	1				
	E		ZIP		5	1				
	E									
	E									
	E									

Searching an Array for a Specific Element

You can search (or look up) an array to determine if a specific element of data is stored in the array. On *the C-(processing) specification,* you define:

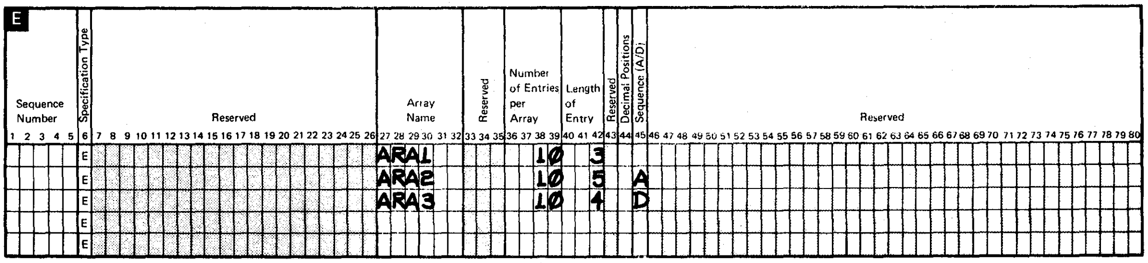


WSU continues the array lookup, one element at a time, beginning with the first element, until the search condition is satisfied or until the end of the array is reached, whichever happens first.

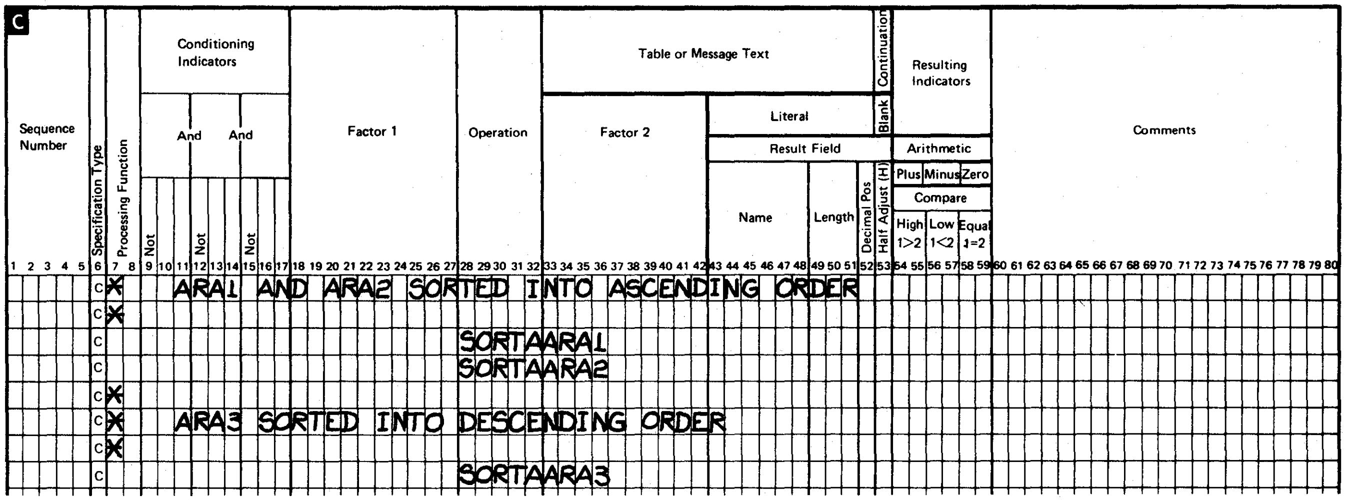
Sorting Array Elements before a Search

If the search is for either a low or a high condition, array elements must be in sequence (ascending or descending). If you are searching an out-of-sequence array for a high/low condition, you must first code a SORTA operation to ensure that the array is in its proper order.

The following examples show the E-specifications and the C-specifications for sorting an array into ascending *A* and descending *D* order. If the sequence is left blank for a SORTA operation, the array is automatically sorted into ascending order.



```
     E                    ARA1       10  3
     E                    ARA2       10  5  A
     E                    ARA3       10  4  D
     E
     E
```



```
     C*   ARA1 AND ARA2 SORTED INTO ASCENDING ORDER
     C*
     C                     SORTAARA1
     C                     SORTAARA2
     C*
     C*   ARA3 SORTED INTO DESCENDING ORDER
     C*
     C                     SORTAARA3
```

Starting the Search at a Particular Element

If you specify just the array name as factor 2 for a LOKUP operation, WSU begins the search at the first element of that array. However, you can also begin an array search at any element you wish by specifying, for factor 2, the array name followed by a comma and an index. As described previously, the index (whether an actual number or the name of a field containing a number) points to the array element where the search is to begin.

In a large array where you know that the value you are searching for is not in a particular section of the array, you can greatly decrease WSU's search time by beginning the lookup at a particular element.

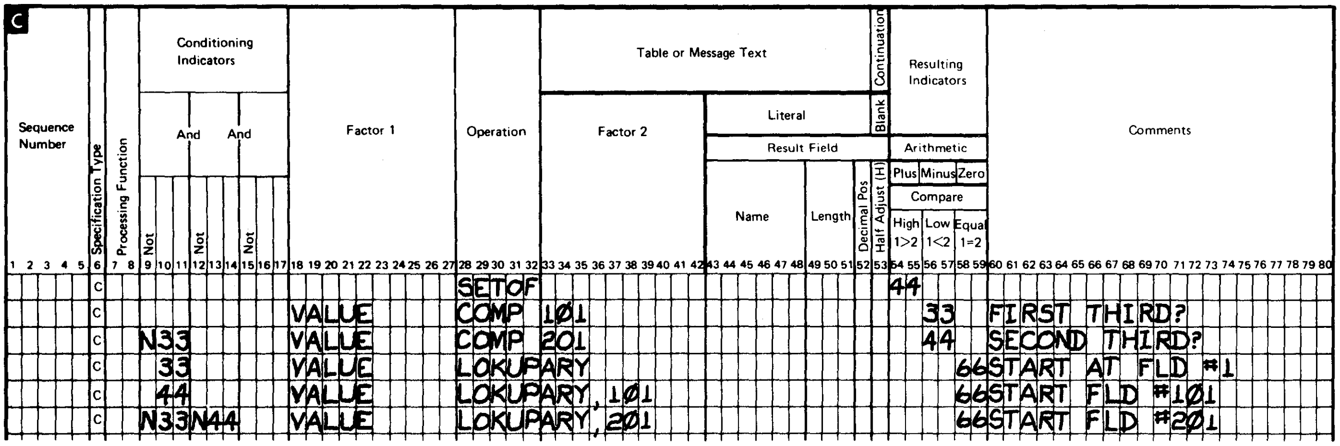
Suppose you have a 300-element array named ARY that contains the values 001 through 300 in ascending sequence. To locate a value of 047, only 47 elements would have to be checked before the search condition was satisfied, but to locate the value 289, 289 elements would have to be checked, *if* the search began at the first array element.

If you divide the array, though, into three parts of 100 elements each, then:

- For any value of less than 101, WSU searches the first third of the array, beginning at element 1.
- For values greater than 100 but less than 201, WSU searches the second third of the array, beginning at element 101.
- For any value greater than 200, WSU searches the last third of the array, beginning at element 201.

In any case, no more than 100 elements have to be checked to satisfy the search condition.

For this example, the number of the array element at which the search is to begin will vary, depending on the value being searched for. Three LOKUP operations are coded, only one of which is done for a particular value:



Specification Type	Conditioning Indicators	Factor 1	Operation	Factor 2	Resulting Indicators High 1>2	Low 1<2	Equal 1=2	Comments
C			SETOF		44			
C		VALUE	COMP	101		33		FIRST THIRD?
C	N33	VALUE	COMP	201		44		SECOND THIRD?
C	33	VALUE	LOKUP	ARY			66	START AT FLD #1
C	44	VALUE	LOKUP	ARY,101			66	START FLD #101
C	N33N44	VALUE	LOKUP	ARY,201			66	START FLD #201

To decide which LOKUP operation is done, you must first find in which part of the array the value is located. The first COMP (compare) operation checks for a value in the first 100 elements. If the value is less than 101, indicating the first one-third of the array, indicator 33 is set on; and if indicator 33 is on, the LOKUP operation beginning at element 1 is done. However, if the value is not in the first third of the array (indicator 33 is not on), another COMP operation is necessary to see if the value is in the second third of the array. If it is there, indicator 44 is set on, and the LOKUP operation that begins at element 101 is done.

If neither indicator 33 nor indicator 44 is set on, the value must be in the last third of the array (if it is in the array at all). Therefore, with both indicators 33 and 44 off, the LOKUP operation that begins at element 201 is done.

For the first LOKUP operation, you need not actually specify the numeric value 1 as the index (as you specify 101 for the second LOKUP) because, you will remember, when no index is specified with the array name, the search automatically begins at the first field, just as if the index were 1.

Note: Setting off indicator 44 (line 01) prevents an error in the lookup function. If the SETOF operation was not used and indicator 44 was set on the the first cycle and indicator 33 in the second cycle, indicator 44 would not be set off in the second cycle because the N33 condition would not be satisfied in line 03. Thus, both lines 04 and 05 would be executed. The LOKUP operation in line 04 would be successful, and indicator 66 would turn on. The LOKUP operation in line 05 would not be successful, and indicator 66 would be turned off. Thus, a not-found condition would result even though the LOKUP was successful.

If the value of the index changes, as in the following example, you can use an index field to contain the number of the array field, rather than using the actual number. In this way, you need code only one LOKUP operation. Of course, you must place the appropriate number in the index field every time before the LOKUP operation is performed, so an index field will not always reduce the number of specification statements required.

C

Sequence Number	Specification Type	Conditioning Indicators	And	And	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Pos	Resulting Indicators High 1>2	Low 1<2	Equal 1=2	Comments
	C					SETOF					44			
	C				VALUE	COMP	101					33		FIRST THIRD?
	C	N33			VALUE	COMP	201					44		NO-SECND THIRD?
	C	33				Z-ADD	001	IXFLD	3	0				START AT FLD #1
	C	44				Z-ADD	101	IXFLD						START FLD #101
	C	N33	N44			Z-ADD	201	IXFLD						START FLD #201
	C				VALUE	LOKUP	ARY,IXFLD						66	

As shown above: First, the COMP operations are done to see whether the value is in the first, second, or last third of the array. Then, the results of the compare operations determine which number should be zero-added into the index field (IXFLD) before the lookup is done.

Determining If a Search is Successful

At this point, we should discuss the index field and how its contents are changed as a result of the LOKUP operation. Before the search is done, you determine the value to be placed in the index field; the array search then begins at the element number specified. The search continues, one element at a time, until the search condition is satisfied or until the end of the array has been reached, whichever happens first.

If *an index field* is specified, WSU puts the number of the first array element that satisfies the search condition into the index field. However, if the end of the array is reached with none of the elements satisfying the search, WSU puts a 1 into the index field.

But if *an actual number* is specified as the index, rather than a field, the actual index is not changed to reflect the success of the search in any case.

You can determine if a search was successful by checking to see whether the resulting indicator assigned has been turned on. Thus, if the resulting indicator is *not on* and an index field has been specified, the index field should contain the value 1, the result of an *unsuccessful* search. If the first field of an array satisfied the search condition, the index field would also contain the value 1; however, in this case, the resulting indicator would be *on*.

Referencing an Element that Satisfies a Search

After a successful search, you can use the data from the element that satisfied the condition only if you specify the array name *with an index field* for the LOKUP operation, because WSU has then placed the number of the field that satisfied the search into the index field. Therefore, specifying the array name with the index field in a later operation refers to the element that satisfied the search.

However, if no index field is available (that is, you specify the array name alone or with a numeric index), you cannot determine the number of the element and, therefore, cannot reference the data. You can only determine *if* one of the array elements does contain the data for which you searched, according to whether the resulting indicator is on.

Remember, for operations that follow an array LOKUP operation, specifying the array name alone refers to the entire array, rather than to any particular element. The only way an individual array element can be referenced is by specifying the array name with an index.

Assume you wish to search an array named CHG to look for amounts over $100. If you only want to determine **if there are any** elements containing a greater amount, the search can be coded like this:

```
     C           100.00    LOKUPCHG
```

If indicator 16 is off, indicating an unsuccessful search, you can then display a message stating that all charges are under or equal to $100.

```
     C           100.00    LOKUPCHG                  16
     C  N16                IMSG 'NO CHARGE OVER $100X
     C
```

With the LOKUP operation shown, however, you would have no way of knowing *how many* elements or *which* elements satisfied the search condition. If you want to know which element satisfied the search or, perhaps, how much over $100 the amount is, the array lookup should be coded with an index field:

```
     C                     Z-ADD1         IX      30
     C*
     C*
     C           100.00    LOKUPCHG,IX
```

The index field can be set to first contain the value 1, so the search begins at the first element of the array. If the search is satisfied, the index field (IX) will contain the number of the first element over $100 and the resulting indicator will be turned on. The contents of field IX can then be displayed to indicate which element satisfied the search.

Searching an Array for More than One Element

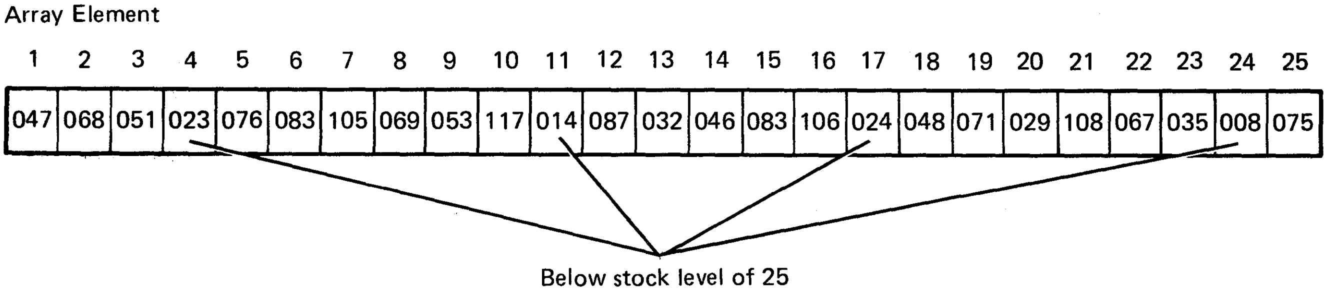
The previous example points out an important consideration: An array LOKUP operation is completed when WSU finds *the first element* that satisfies the search condition. If you wish to find *all elements* that satisfy the condition, you must code additional specifications to cause the program to *loop back* to repeat the lookup operation from the point where the last search was successful.

As an example, assume your company manufactures 25 different items, identified by item codes 1 through 25. You use a 25-element array named QTY to keep track of the quantity in stock of each item: the first element contains the quantity of item code 1, the second element contains the quantity of item code 2, and so on. Whenever the quantity of an item falls below 25, the plant is to produce 100 more of that item and add them to stock.

To determine which items are to be manufactured, your program searches the QTY array every week, comparing the array elements with a search word, MFGPT, a field that contains a value of 25. When a quantity is found to be less than 25 (that is, the search condition is *low)*, the item code and quantity in stock are displayed, or perhaps written to a transaction file.

You can see that four items must be manufactured this week:

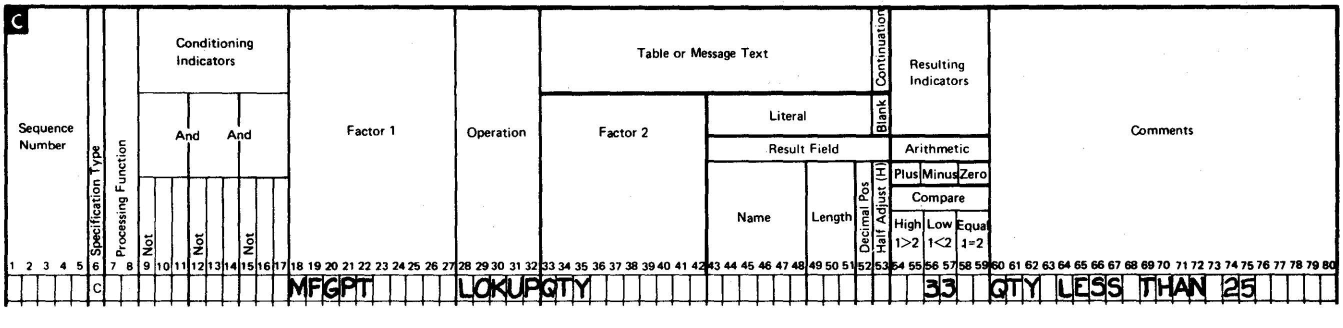
QTY ARRAY



Array Element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	047	068	051	023	076	083	105	069	053	117	014	087	032	046	083	106	024	048	071	029	108	067	035	008	075

Below stock level of 25

The following C-specification will not locate *all* of the items with quantities less than 25; rather, this LOKUP operation will locate only *the first* quantity below 25:



Sequence Number	Specification Type	Factor 1	Operation	Factor 2	Resulting Indicators High 1>2	Resulting Indicators Low 1<2	Resulting Indicators Equal 1=2	Comments
	C	MFGPT	LOKUP	QTY		33		QTY LESS THAN 25

To locate more than one element satisfying the same search condition, you must repeat the LOKUP operation, starting the search at the point where the previous search ended. You can use the GOTO and TAG operations to repeat the lookup. To make sure the repeated search begins where the last search left off, you must specify the array name *with an index field* for the LOKUP operation. The first search should begin at element 1; the contents of the index field are then updated after each successful search to indicate at which array element the next search should begin:

Specification Type	Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Pos	Resulting Indicators High 1>2	Low 1<2	Equal 1=2	Comments
C			Z-ADD	01	IX	2	0				START AT FLD #1
C		AGAIN	TAG								
C			SETOF					33	44		
C		MFGPT	LOKUP	QTY,IX					33		QTY LESS THAN 25
C	33		PUT	FILENAME							
C	33		SETOF					33			
C		IX	ADD	01	IX						ADD TO INDEX
C		IX	COMP	26					44		END OF ARRAY?
C	44		GOTO	AGAIN							NO-SEARCH NEXT

The index field (IX) is first set up to contain the value 1 (the field is zeroed before adding 1 because you have no way of knowing the contents of IX at the beginning of the program). The TAG operation is not done, so WSU skips this statement and does the LOKUP operation.

When the first QTY element less than 25 is found, the number of the element (04) is placed in the index field. Providing the LOKUP operation was successful (that is, indicator 33 is on), the results are written to a file. A 1 is then added to the value in the index field, to indicate at which element the next search should begin. The value in the index field is then compared to 26 to see if the entire array (25 fields) has been searched. If there are still array elements to be checked (that is, indicator 44 is on), the program branches back to do the LOKUP again. The search would then begin again, only at the element following the last element that satisfied the search. These specifications would be repeated over and over until all items to be manufactured are located and until the end of the array is reached.

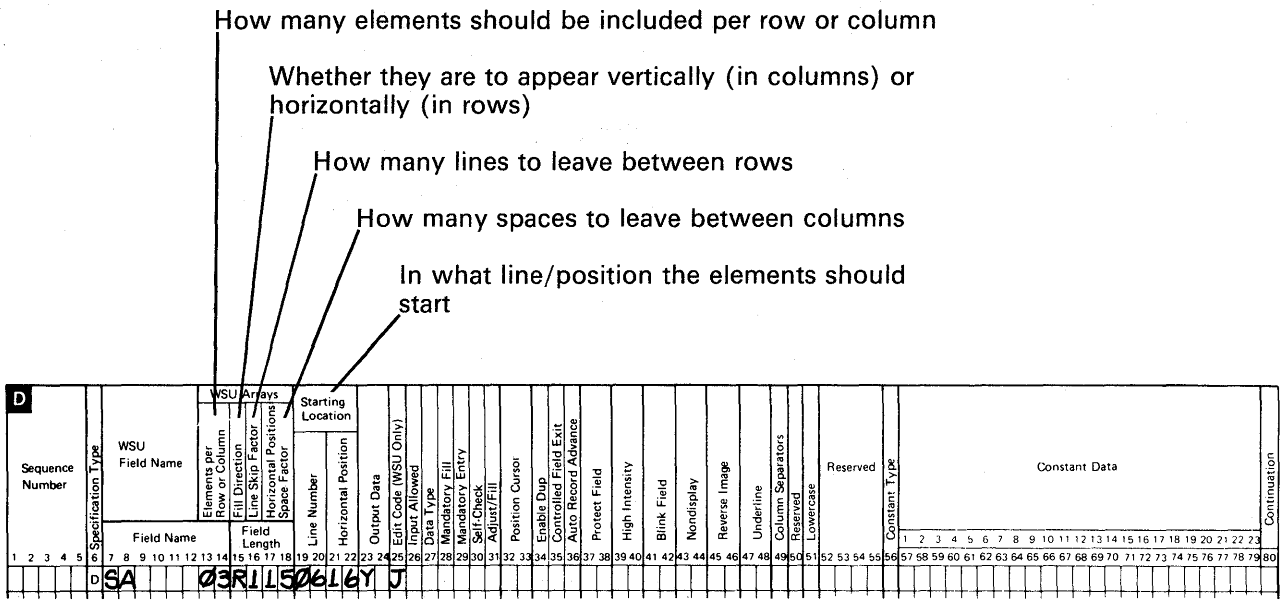
Recording the Results of a Search

The specifications just described search through the QTY array to locate more than one element. In this case, it does no good to search through an array unless you know what data was found; therefore, you could record the results by displaying each quantity less than 25 and its related item code or by writing the data into a transaction file. Following each successful search, the item code number (same as the number of the array element containing the quantity) is stored in the index field IX, and the field IX can be recorded. *Displaying Arrays* later in this chapter describes how to display individual array elements.

Displaying Entire Arrays

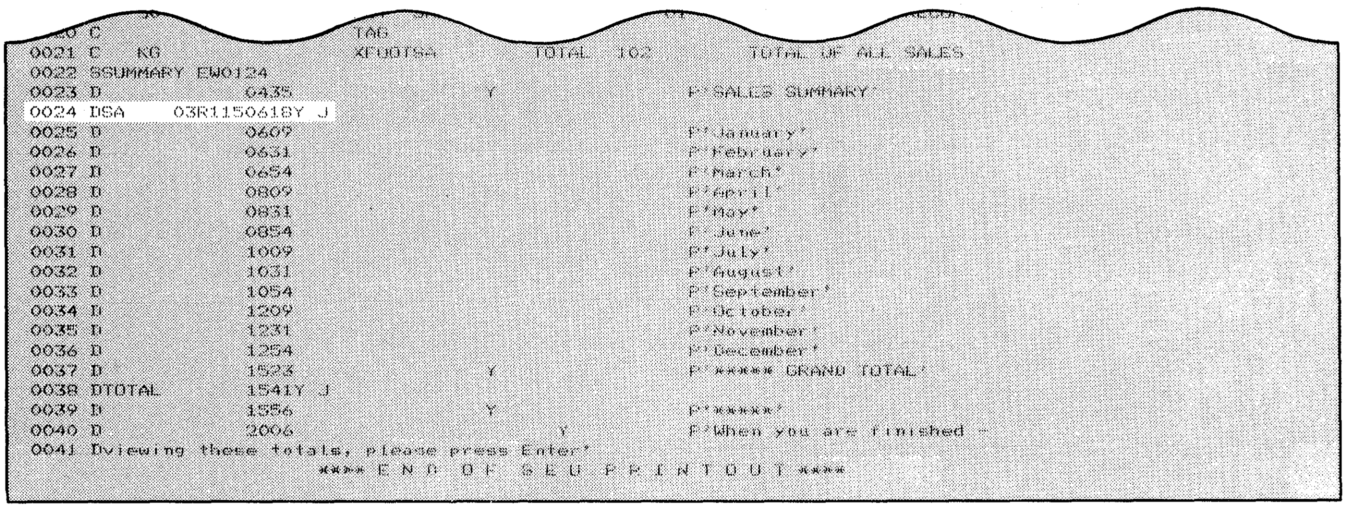
Perhaps you want to look at the contents of an array at some point while the program is running or at the end of the job. An entire array is easily displayed with D-specifications. You specify:

- *Where* you want the data elements to appear on the display:



- *How* you want the data elements to appear on the display; that is, if you want numeric array elements to be edited with commas, decimal points, and so on. (Remember, editing affects the spacing of the elements.)

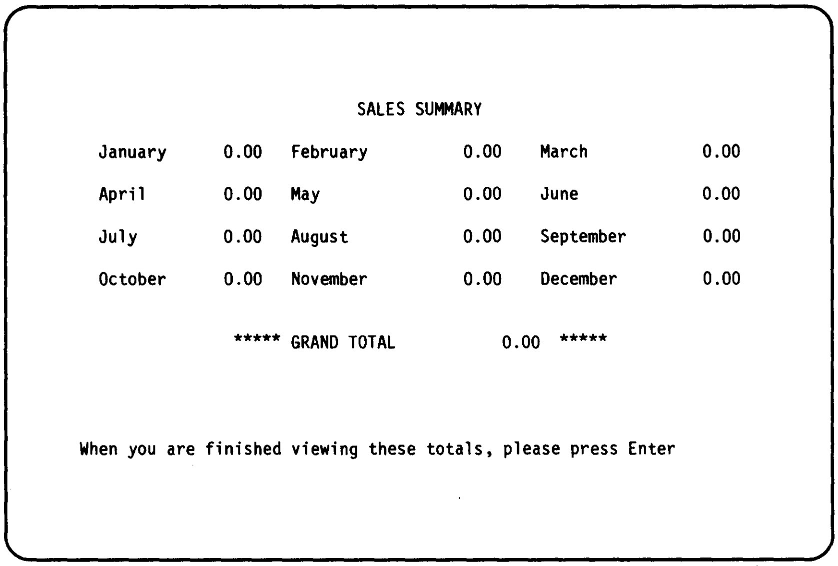
The above D-specification is from the program shown earlier in Figure 5-3; the array is displayed with only one coded D-specification statement. WSU then uses that statement to generate instructions for each element of the array:



```
   0 C                     TAG
0021 C   KG                XFOOTSA        TOTAL   102      TOTAL OF ALL SALES
0022 SSUMMARY EW0124
0023 D            0435                Y               F'SALES SUMMARY'
0024 DSA    03R1150618Y J
0025 D            0609                                F'January'
0026 D            0631                                F'February'
0027 D            0654                                F'March'
0028 D            0809                                F'April'
0029 D            0831                                F'May'
0030 D            0854                                F'June'
0031 D            1009                                F'July'
0032 D            1031                                F'August'
0033 D            1054                                F'September'
0034 D            1209                                F'October'
0035 D            1231                                F'November'
0036 D            1254                                F'December'
0037 D            1523                Y               F'***** GRAND TOTAL'
0038 DTOTAL       1541Y J
0039 D            1556                Y               F'*****'
0040 D            2006                      Y         F'When you are finished -
0041 Dviewing these totals, please press Enter'
                        **** E N D  O F  S E U  P R I N T O U T ****
```

The constants to identify the elements on the display have to be coded separately, of course. However, no C-specifications are required in this example.

The array display looks like this:



```
                              SALES SUMMARY

     January     0.00  February         0.00   March           0.00

     April       0.00  May              0.00   June            0.00

     July        0.00  August           0.00   September       0.00

     October     0.00  November         0.00   December        0.00

                 ***** GRAND TOTAL          0.00  *****



   When you are finished viewing these totals, please press Enter
```

Displaying Individual Elements of an Array

To display individual array elements, you must first move those elements to uniquely named fields and then display the fields. (WSU allows you to index fields only on the C-specification, not on the D-specification.)

In the earlier section on *Searching an Array for More than One Element*, you saw an example of an array being searched for all the items with a quantity less than 25. To record the results of that search you might, each time a match was found, move the quantity in the IX field to a temporary array; then, at the end of the program, move the elements in the temporary array to temporary fields and display the fields.

Loading Arrays

In the beginning of this chapter, you learned that data in an execution-time array such as WSU uses can be loaded in either of the following ways:

- Read from an input file when the program starts
- Created by processing done while the program is running
- Keyed by an operator.

The sections in this chapter on *Referencing Arrays* and *Referencing Individual Array Elements* give examples of loading arrays during processing; this section will discuss ways of loading an array at the beginning of a job.

Reading Input File Data into an Array

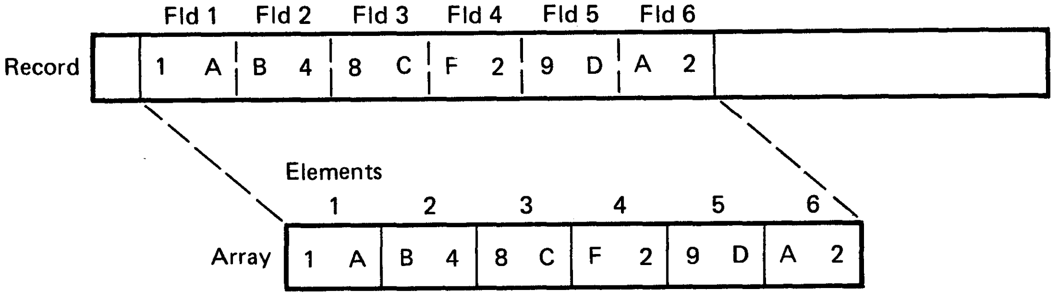
When you define *the array* on the E-specification, WSU sets up an array in storage, ready to receive array data. You must also describe *the input data* to be read into the array on I-specifications, and read the necessary data into the array using C-specifications. The array must be loaded before your program can do any operations that use the array data, so you will probably load the array during IJ or IW processing.

Fields of array data to be read from input records must be described on the I-specifications, which indicate where the data is located on the record. How the array information is described and stored depends on:

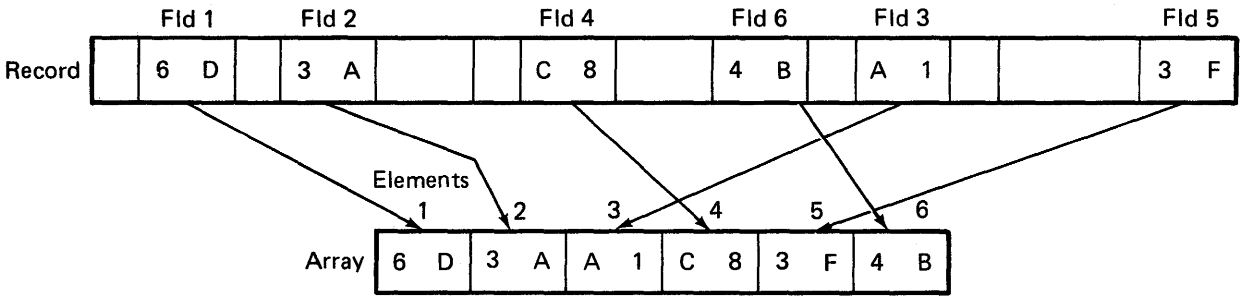
- How the array data is organized on a record
- Whether the array data is contained in one or more records.

An input record containing array data may contain only data for that array or may contain both array data and other data fields to be used in the program. In either case, the array data is organized in one of these ways:

- All array elements may occupy *consecutive positions* on the record; that is, each element immediately following another with no blanks or other data between the elements.



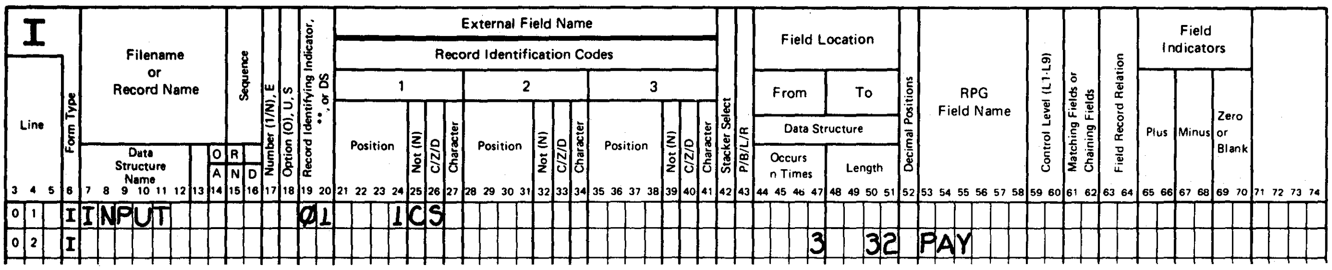
- The array elements may be **scattered** on the record in any order, with blanks or other input fields placed between the array elements.



The way in which the data is organized and the size of the array generally determine the number of input records required to contain the array data.

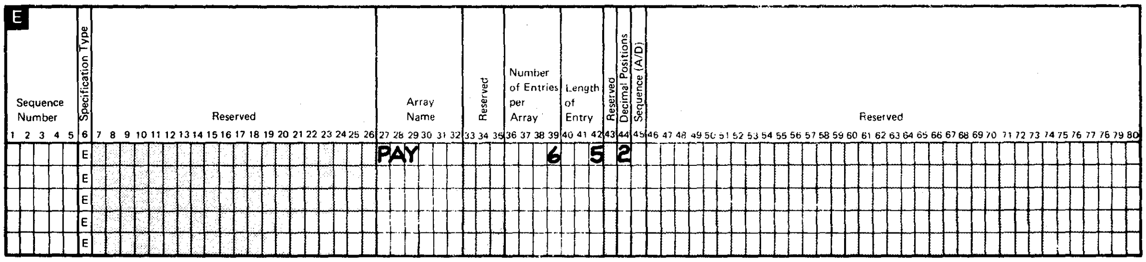
Reading Data from Consecutive Positions

If array elements are in order in consecutive positions on a record, describing and storing the data is very easy. All of the array data on the one record may be described on the I-specification as if it were a single field. Thus, only one I-specification is necessary to indicate a name for the field and the positions in the record where the array data begins and ends:



```
  01 IINPUT       01   1 CS
  02 I                                        3  32 PAY
```

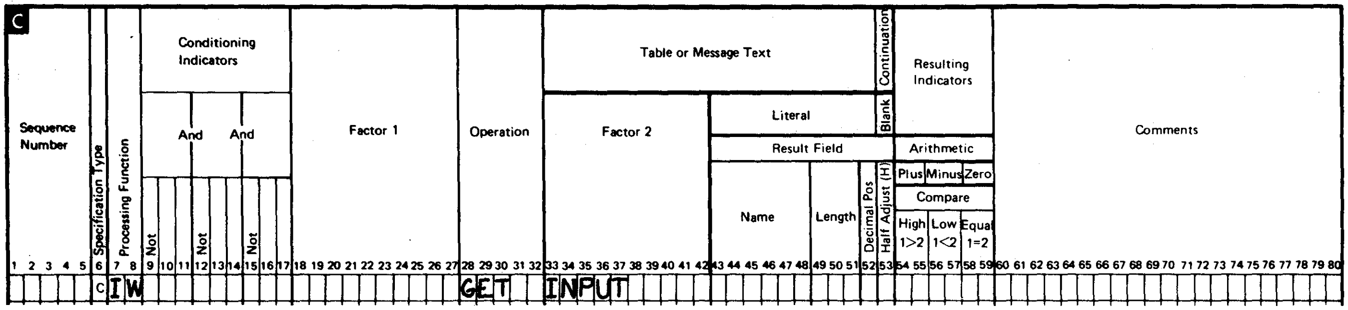
When you describe an input record of array data, you specify no decimal positions, because the description on the E-specification has already done so.



```
     E                    PAY         6  5 2
```

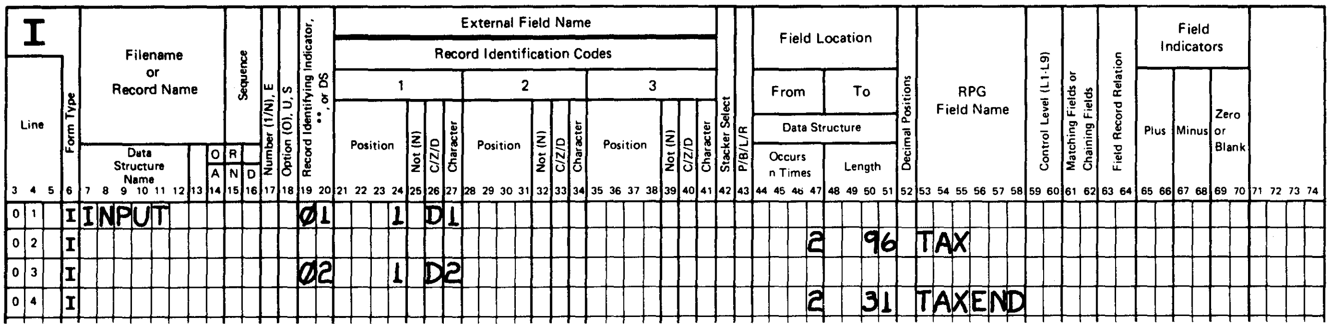
By specifying the name of the array as the result field name, the data is automatically stored in the appropriate elements of the array as the input record is read.

The C-specifications to read consecutive data into an array consist simply of a GET operation to read the record.

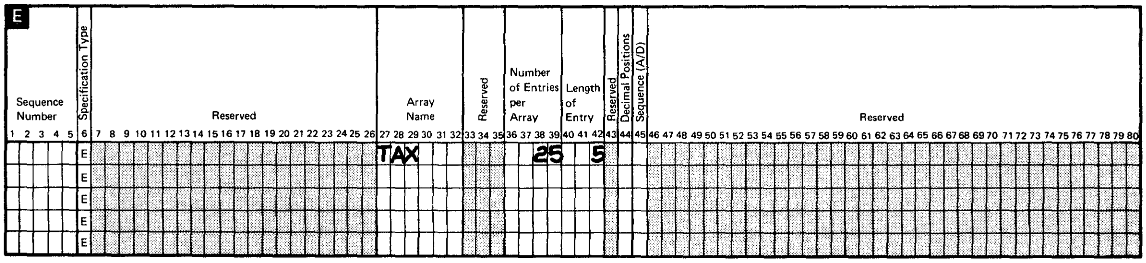


```
     CIW                   GET  INPUT
```

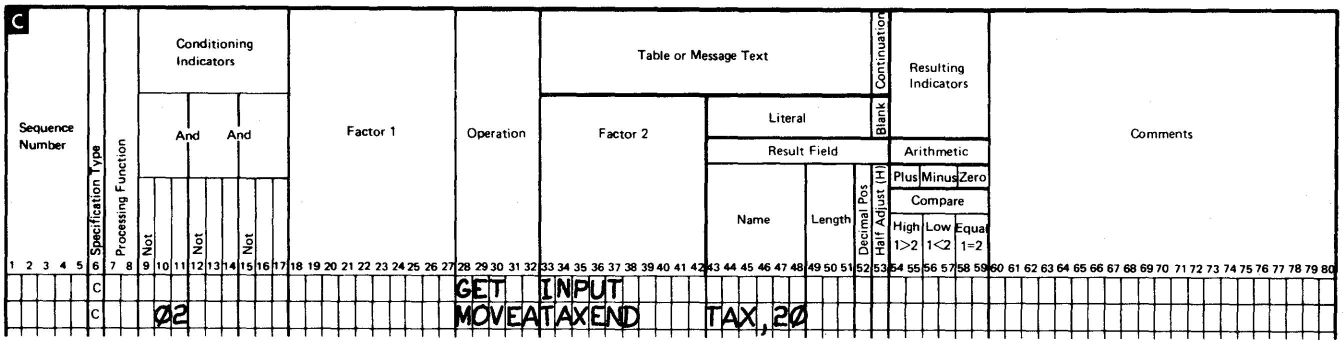
In the preceding example, the array data was on one record. In the following example the array data is organized consecutively on two records. The array data on the second record can also be described as a single field; you would use a MOVEA operation to move data from that field to the array. This is the coding necessary to load the TAX array:



```
  01 IINPUT       01   1 D1
  02 I                                         2  96 TAX
  03 I            02   1 D2
  04 I                                         2  31 TAXEND
```



```
     E                    TAX        25  5
```

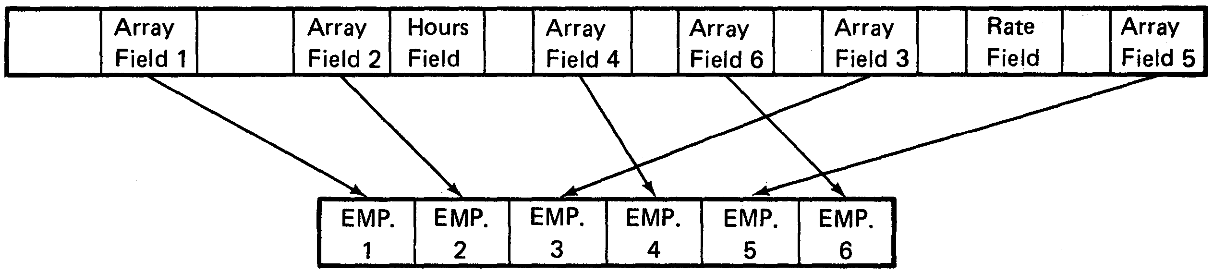


```
     C                     GET  INPUT
     C   02                MOVEATAXEND    TAX,20
```

Reading Data from Scattered Positions

When array elements are scattered on an input record, each field must be described separately on the I-specification to indicate where each item of array data begins and ends. You assign a unique field name to each field of array data on the input record, then code processing operations to move each data field individually into the appropriate array element. (WSU does not allow you to index fields on the I-specifications).

Assume that a six-field array named EMP is set up by coding an E-specification. The six fields of data for the array are scattered on a record; additional input information (blanks and other input fields) exists between the fields. And, the fields are not in the order in which they are to be stored:



When you describe the array data, you must identify each field by a unique field name on separate I-specification statements, because the array data is not continuous. Normal input fields can be described along with the array fields.

Once the scattered fields have been described on the I-specification, each field of array data is stored in the array using a Z-ADD operation. Because each field has a unique field name and must be stored in a specific array element, a separate Z-ADD operation must be coded for each field to be stored.

Regardless of how many records are used to contain array data, if the fields are scattered on the records, each field must be individually loaded into its appropriate position in the array.

```
  01 IINPUT   AA  01   1 D8
  02 I                                        5   90AFLD1
  03 I                                       13  170AFLD2
  04 I                                       18  211HRS
  05 I                                       25  290AFLD4
  06 I                                       35  390AFLD6
  07 I                                       44  480AFLD3
  08 I                                       71  772RATE
  09 I                                       85  890AFLD5
  10 I
  11 I
```

AFLD1, AFLD2: Fields containing array data.

HRS: Normal input field.

AFLD4, AFLD6, AFLD3: Fields containing array data.

RATE: Normal input field.

AFLD5: Field containing array data.

```
     E                    EMP         6  5 0
     E
     E
     E
     E
```

```
     C                      GET  INPUT
     C  N01                 GOTO SKIP
     C                      Z-ADDAFLD1     EMP,1
     C                      Z-ADDAFLD2     EMP,2
     C                      Z-ADDAFLD3     EMP,3
     C                      Z-ADDAFLD4     EMP,4
     C                      Z-ADDAFLD5     EMP,5
     C                      Z-ADDAFLD6     EMP,6
     C           SKIP       TAG
```

However, a separate specification statement is not always necessary for each field of data to be loaded. In some cases, the same statement can be used for all the records, depending on whether:

- All the input records for a single array are organized in the same format
- The fields from different records can be assigned the same name.

Assume that a 22-element array (ARA) is defined. The data for the array is scattered on six input records:

Record 1		FLD1		FLD2		FLD3		FLD4
2		FLD5		FLD6		FLD7		FLD8
3		FLD9		FLD10		FLD11		FLD12
4		FLD13		FLD14		FLD15		FLD16
5		FLD17		FLD18		FLD19		FLD20
6		FLD21		FLD22				

Although the array data is not consecutive, the four fields on each of the first five records are the same format on each record; the remaining two fields on the sixth record are in the same format as the first two fields on all other records.

Because the array data follows the same organization on all records, describing one set of fields actually describes the fields on all records, except the last. A separate I-specification should be coded to indicate that record 6 has only two of the fields:

Line	Form Type	Filename or Record Name	A N D / O R	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Field Location From	Field Location To	Decimal Positions	RPG Field Name
01	I	INPUT		01	1		D	1				
02	I		OR	02	1		D	2				
03	I		OR	03	1		D	3				
04	I		OR	04	1		D	4				
05	I		OR	05	1		D	5				
06	I								5	9	0	FLDA
07	I								23	27	0	FLDB
08	I								41	45	0	FLDC
09	I								76	80	0	FLDD
10	I			06	1		D	6				
11	I								5	9	0	FLDA
12	I								23	27	0	FLDB

Note that records 1 through 5 are described in an OR relationship. You can assume that the array input records are in sequence, where record type 1 is the first record read and record type 6 is the last.

Because the fields on the different records have the same field names, only one MOVE operation is necessary for each unique field name. The first MOVE operation, when repeated for each record, moves field FLDA of that record to the appropriate element of the ARA array. The last two moves are done for every record except the last, which does not have fields FLDC and FLDD.

The fields on the input records are in the same order as they are to be stored in ARA, so a definite pattern is established as to where the data is to be moved: fields from record 1 are stored in array elements 1 through 4, fields from record 2 in elements 5 through 8, and so on.

C

Sequence Number 1-5	Specification Type 6	Processing Function 7-8	Conditioning Indicators Not 9	10-11	Not 12	13-14	Not 15	16-17	Factor 1 18-27	Operation 28-32	Factor 2 33-42	Result Field Name 43-48	Length 49-51	Decimal Pos 52	Half Adjust (H) 53	Resulting Indicators Plus High 1>2 54-55	Minus Low 1<2 56-57	Zero Equal 1=2 58-59	Comments 60-80
	C			01						Z-ADD	1	A	2	0					SET UP INDEX
	C			01						Z-ADD	2	B	2	0					FIELDS FOR
	C			01						Z-ADD	3	C	2	0					MOVING DATA
	C			01						Z-ADD	4	D	2	0					
	C	*																	
	C	*																	
	C									MOVE	FLDA	ARA,A							MOVE DATA TO
	C									MOVE	FLDB	ARA,B							PROPER ARRAY
	C		N	06						MOVE	FLDC	ARA,C							FIELDS
	C		N	06						MOVE	FLDD	ARA,D							
	C	*																	
	C		N	06					A	ADD	4	A							INCREASE INDEX
	C		N	06					B	ADD	4	B							VALUES TO MOVE
	C		N	06					C	ADD	4	C							DATA FROM NEXT
	C		N	06					D	ADD	4	D							RECORD

Summary

This chapter has shown you ways in which to:

- Define
- Reference
- Search
- Display
- Load.

execution-time arrays in your WSU programs. For specific coding information, refer to:

- Chapter 11, which contains an array program
- Chapter 14, which contains coding of the E-specification
- Chapter 16, which contains coding of the D-specification
- Chapter 17, which contains coding of the C-specification.

Chapter 6. Coding Displays

The S-(display control) specifications describe the characteristics of an entire display; whereas the D-(field definition) specifications are coded to describe the attributes or characteristics of each of the fields on the display. Displays are coded using the *Display Format Specifications Coding Sheet* (Figure 6-1).

This chapter includes a description of the entries you can code on the S- and D-specifications.

The term *this display* refers to the display you are currently describing on the S- and D-specifications.

For examples of coded S- and D-specifications, refer to Chapter 11, *WSU Example Programs*. For detailed information on the possible entries on the coding form, see Chapter 16, *S-(Display Control) and D-(Field Definition) Specifications*. More about the coding and designing of displays can be found in the manual, *Creating Displays*.

Guidelines for Planning a Display

One of two methods can be used to plan displays and create S- and D-specifications for them. One method is to use display layout sheets for drawing displays before coding them on the S- and D-specifications. The other method is SDA, which may be faster. Refer to the manual, *Creating Displays*, for both methods.

System/36 Display Format Specifications

Display Control Specification

Sequence Number	Specification Type	Display Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Null Fill	Reserved	Right-to-Left Display	WSU Only													Reserved	Key Mask	Reserved
																			Enter Mode Sequence							Review Mode Record Identifying Indicators			Insert Mode Record Identifying Indicators					
																			Start	End	Entry Required	Repeat	Reserved	Priority	Preprocess	1	2	3	1	2	3			
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15 16	17 18	19 20	21	22	23 24	25 26	27	28	29 30	31 32	33 34	35 36	37 38	39	40	41	42	43	44	45	46	47	48 49	50 51	52 53	54 55	56 57	58 59	60 61 62 63	64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
	S																																	

The "H" specifications are optional. If you choose to use the "H" specifications, they must be placed between the "S" specifications and the "D" specifications for a display format. Use the other side of this form to code the "H" specifications.

Field Definition Specification

Sequence Number	Specification Type	WSU Field Name	WSU Arrays				Starting Location		Output Data	Edit Code (WSU Only)	Input Allowed	Data Type	Mandatory Fill	Mandatory Entry	Self-Check	Adjust/Fill	Position Cursor	Enable Dup	Controlled Field Exit	Auto Record Advance	Protect Field	High Intensity	Blink Field	Nondisplay	Reverse Image	Underline	Column Separators	Reserved	Lowercase	Reserved	Constant Type	Constant Data	Continuation
			Elements per Row or Column	Fill Direction	Line Skip Factor	Horizontal Positions Space Factor	Line Number	Horizontal Position																									
		Field Name		Field Length																												1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	
1 2 3 4 5	6	7 8 9 10 11 12	13 14	15	16	17 18	19 20	21 22	23 24	25	26	27	28	29	30	31	32 33	34	35	36	37 38	39 40	41 42	43 44	45 46	47 48	49	50	51	52 53 54 55	56	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																
	D																																

Figure 6-1. System/36 Display Format Specifications

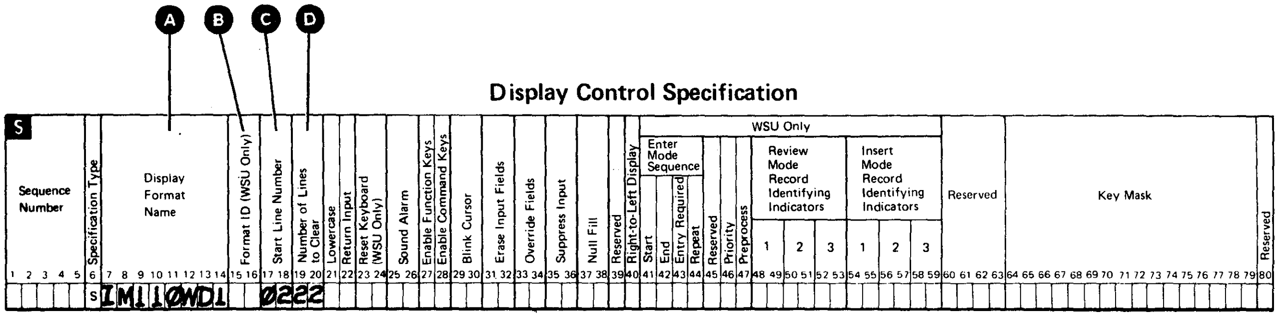
S-(DISPLAY CONTROL) SPECIFICATION

Identification

The *Display Format Name* (columns 7 through 14) is the only required entry on the S-specification. The name that you give the display that WSU creates from the S- and D-specifications can be any name you wish as long as it fits the guidelines specified in Chapter 16, *S- and D-Specifications.* Ⓐ

The 2-character *Format ID* (columns 15 and 16) allows an operator to select this display from the WSU display. You can also use indicators (see Chapter 3, *Reserved Fields and Indicators* to put restrictions on when this display can be used. Ⓑ

You can specify the exact starting line and position *(Start Line Number)* (columns 17 and 18) in which you want your display to start and how many lines to clear *(Number of Lines to Clear)* (columns 19 and 20) before you start the display. Ⓒ Ⓓ

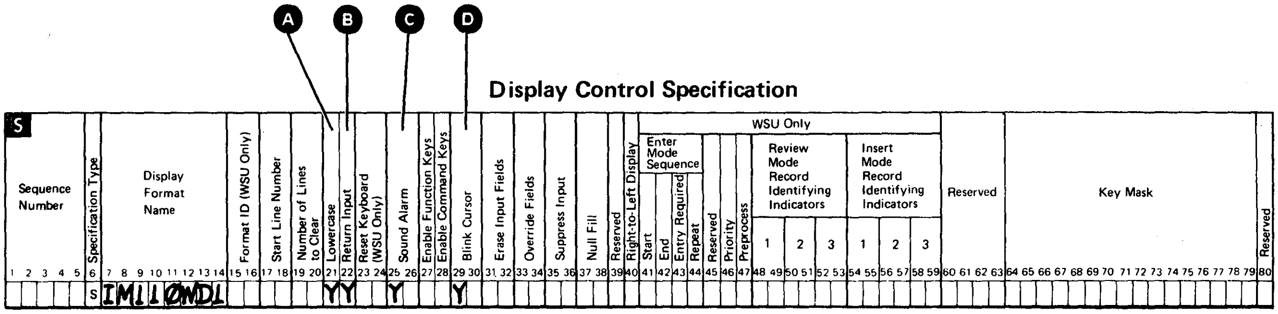

Display Control Specification

S IM110WD1 0222

Attributes

You can specify the following attributes for your display by entering a positive (Y) response for:

- *Lowercase* (column 21) to have your input fields in lowercase or uppercase and lowercase (by using the Shift key). Ⓐ
- *Return Input* (column 22) means you should have input or output/input fields on your display. Ⓑ
- *Sound Alarm* (columns 25 and 26) causes an alarm to sound when your display appears. Ⓒ
- *Blink Cursor* (columns 29 and 30) causes the cursor to blink when the display appears or when certain conditions (that you set) happen when the display appears. Ⓓ



Display Control Specification

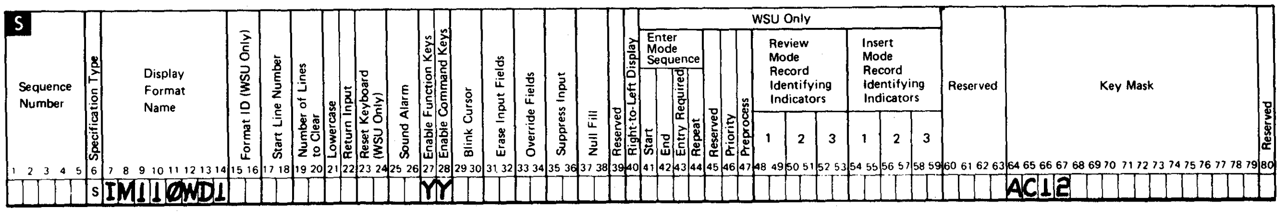
Sequence Number 1-5	Specification Type 6	Display Format Name 7-14	Format ID (WSU Only) 15-16	Start Line Number 17-18	Number of Lines to Clear 19-20	Lowercase 21	Return Input 22	Reset Keyboard (WSU Only) 23-24	Sound Alarm 25-26	Enable Function Keys 27	Enable Command Keys 28	Blink Cursor 29-30	Erase Input Fields 31-32	Override Fields 33-34	Suppress Input 35-36	Null Fill 37-38	Reserved 39	Right-to-Left Display 40	WSU Only: Enter Mode Sequence Start 41	End 42	Entry Required 43	Repeat 44	Reserved 45	Priority 46	Preprocess 47	Review Mode Record Identifying Indicators 1 48-49	2 50-51	3 52-53	Insert Mode Record Identifying Indicators 1 54-55	2 56-57	3 58-59	Reserved 60-63	Key Mask 64-79	Reserved 80
	S	IM110WD1				Y	Y		Y			Y																						

Function/Command Keys and Key Mask

You can use *Enable Function Keys* (column 27) and *Enable Command Keys* (column 28) to control your program's operations. A Y in these fields enables the keys. The *Key Mask* (columns 64 through 79) identifies the command and function keys that you want enabled or disabled when this display appears.

By not enabling these keys, you prevent operators from pressing keys by accident and perhaps ending up with results that are not predictable.

Display Control Specification



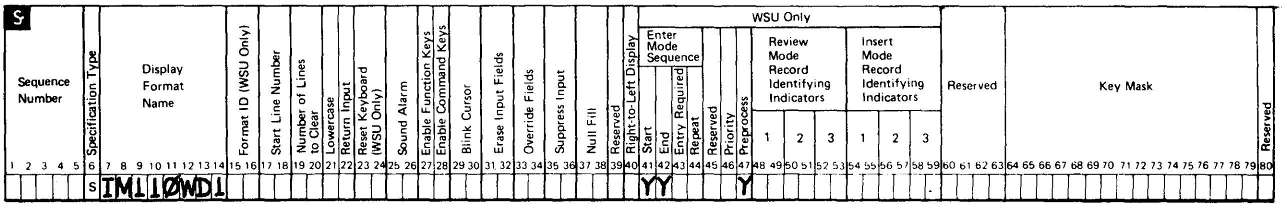
Sequence Number 1-5	Specification Type 6	Display Format Name 7-14	Format ID (WSU Only) 15-16	Start Line Number 17-18	Number of Lines to Clear 19-20	Lowercase 21	Return Input 22	Reset Keyboard (WSU Only) 23-24	Sound Alarm 25-26	Enable Function Keys 27	Enable Command Keys 28	Blink Cursor 29-30	Erase Input Fields 31-32	Override Fields 33-34	Suppress Input 35-36	Null Fill 37-38	Reserved 39	Right-to-Left Display 40	WSU Only: Enter Mode Sequence Start 41	End 42	Entry Required 43	Repeat 44	Reserved 45	Priority 46	Preprocess 47	Review Mode Record Identifying Indicators 1 48-49	2 50-51	3 52-53	Insert Mode Record Identifying Indicators 1 54-55	2 56-57	3 58-59	Reserved 60-63	Key Mask 64-79	Reserved 80
	S	IM110WD1								Y	Y																						AC12	

Display Sequences

A sequence of displays consists of one of more displays which you code in the order that they should appear during an enter mode sequence cycle. IJ, IW, EW, and EJ displays cannot be sequenced.

Entries for *Enter-Mode Sequence* (columns 41 through 44) (see Chapter 2 for a discussion of the WSU modes), indicate if this display is first in a sequence of displays, if this display is the last in a sequence of displays, if you want to require an entry on the display, and if you want to repeat this display to accept input until the operator keys all of the information needed. These columns are ignored if the application program is not in enter mode.

Display Control Specification



S																			WSU Only															
Sequence Number	Specification Type	Display Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Null Fill	Reserved	Right-to-Left Display	Enter Mode Sequence				Reserved	Priority	Preprocess	Review Mode Record Identifying Indicators			Insert Mode Record Identifying Indicators			Reserved	Key Mask	Reserved
																			Start	End	Entry Required	Repeat				1	2	3	1	2	3			
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15 16	17 18	19 20	21	22	23 24	25 26	27	28	29 30	31 32	33 34	35 36	37 38	39	40	41	42	43	44	45	46	47	48 49	50 51	52 53	54 55	56 57	58 59	60 61 62 63	64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
	S	IM110WD1																	Y	Y					Y						.			

You can indicate, on the S-specification, if this display:

- Is the first in a sequence (Y *Start* field)
- Is the last in a sequence (Y *End* field)
- Is in the middle of a sequence (NN or blanks in *Start* and *End* fields)
- Is a sequence of only one display (YY in *Start* and *End* fields)
- Requires an entry in an input field (Y in *Entry Required* field)
- Should be repeated (Y *Repeat* field).

You should have an end display for each start display in a sequence. If the first display that follows IJ and IW displays is not specified as a start display, WSU assumes that this display is the start of the primary sequence.

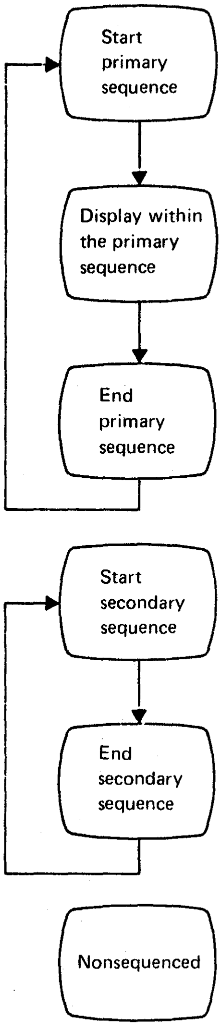
If you code two start displays without an end display between them, WSU ignores the start entry for the second display.

If you code two end displays without a start display between them, WSU ignores the second end display and treats the displays that follow the first end display as nonsequenced displays.

If you code neither a start display nor an end display in your program, WSU assumes that the first display that follows the IJ and IW is the start of a primary sequence and that the last display that comes before the EW and EJ displays is the end of the primary sequence.

If you code a start display but do not code an end display, WSU assumes that the last display in the program (other than the EW and EJ displays) is an end display.

See Figure 6-2 for entries in the Start and End columns for sequenced and nonsequenced data.



	41 (Start)	42 (End)
Start primary sequence	Y	blank
Display within the primary sequence	blank	blank
End primary sequence	blank	Y
Start secondary sequence	Y	blank
End secondary sequence	blank	Y
Nonsequenced	blank	blank

Figure 6-2. Entries in Start/End for Sequenced and Nonsequenced Data

Priority

You can specify the expected frequency of the use of your displays by coding a 0, 1, 2, or 3 in column 46. An entry of 3 represents the highest priority. The other priority entries, from high to low usage, are 2, 1, and 0. Refer to *Appendix B* for a discussion of assigning priority to display format processing.

Preprocess

By entering a Y in the *Preprocess* (column 47) field of the S-specification, you can specify that some processing for your display occur before the display is shown. See Figure 6-3 for a flowchart of preprocessing logic. Preprocessing is described in Chapter 7.

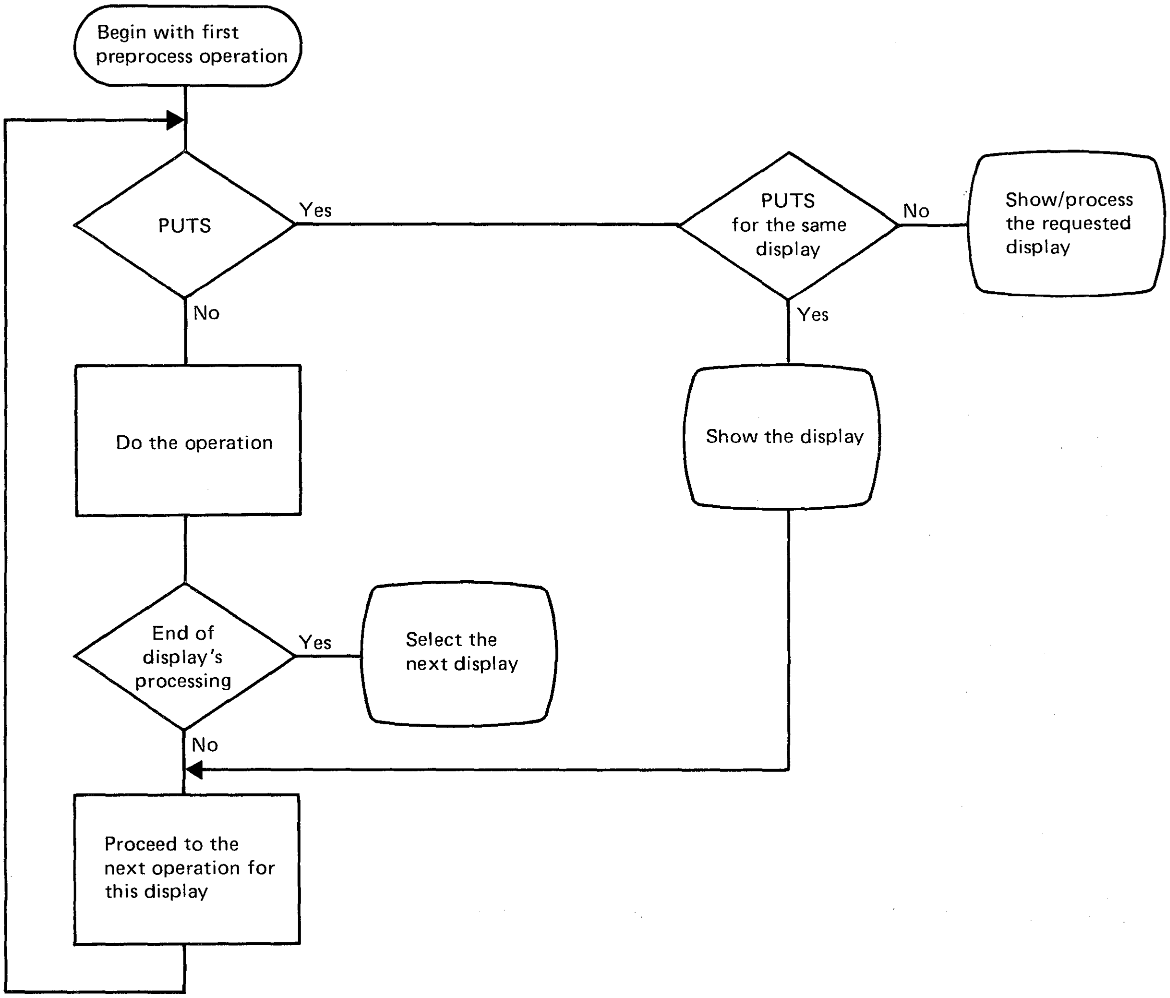
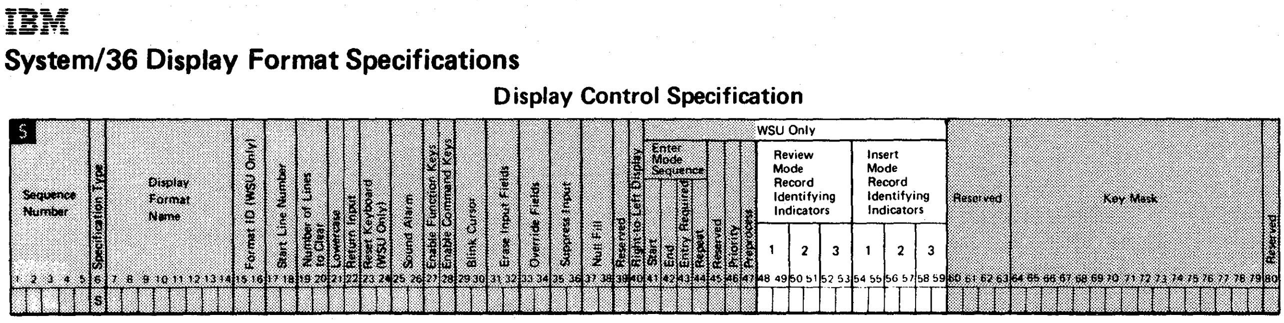


Figure 6-3. Preprocessing Logic

Record Identifying Indicators

The record identifying indicators you specify on the S-specification correspond to those on the I-specification of the file definition.

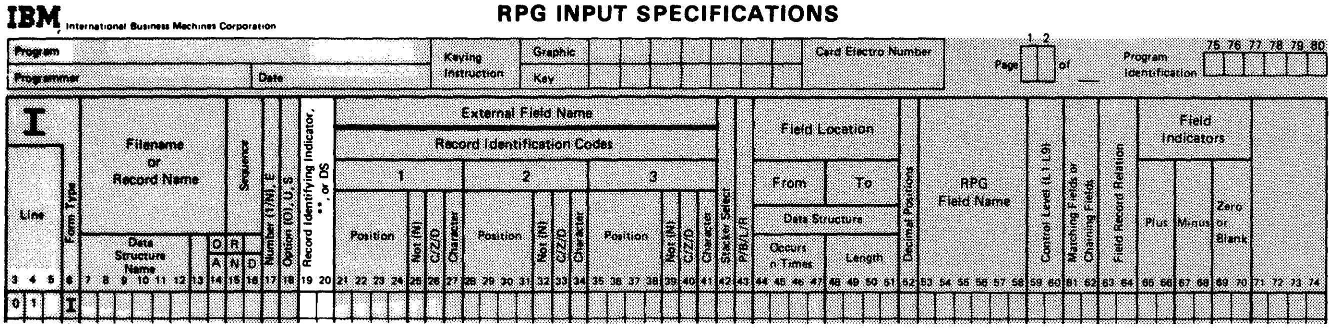


IBM

System/36 Display Format Specifications

Display Control Specification

																			WSU Only															
																			Enter Mode Sequence							Review Mode Record Identifying Indicators			Insert Mode Record Identifying Indicators					
Sequence Number	Specification Type	Display Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Null Fill	Reserved	Right-to-Left Display	Start	End	Entry Required	Repeat	Reserved	Priority	Preprocess	1	2	3	1	2	3	Reserved	Key Mask	Reserved
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15 16	17 18	19 20	21	22	23 24	25 26	27	28	29 30	31 32	33 34	35 36	37 38	39	40	41	42	43	44	45	46	47	48 49	50 51	52 53	54 55	56 57	58 59	60 61 62 63	64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
	S																																	



IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Program		Keying Instruction	Graphic	Card Electro Number
Programmer	Date		Key	

Page 1 2 of ___ Program Identification 75 76 77 78 79 80

I									External Field Name																								
									Record Identification Codes														Field Location							Field Indicators			
									1				2				3						From	To									
																							Data Structure										
Line	Form Type	Filename or Record Name Data Structure Name		O R A N D	Sequence	Number (1/N), E	Option (O), U, S	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	Occurs n Times	Length	Decimal Positions	RPG Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank	
3 4 5	6	7 8 9 10 11 12	13	14 15 16		17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38	39	40	41	42	43	44 45 46 47	48 49 50 51	52	53 54 55 56 57 58	59 60	61 62	63 64	65 66	67 68	69 70	71 72 73 74
0 1	I																																

By specifying one or more *Review Mode Record Identifying Indicators* (columns 48 through 53), you designate this display as *review-capable* for use in review mode, and you indicate that only the transaction file record type(s) specified can be reviewed using this display.

By also specifying indicators for *Insert Mode* (columns 54 through 59), you indicate that records can only be inserted after the specified record type(s). Note that to allow insert mode, you must specify record identifying indicators for *both* review mode and insert mode.

See Chapter 2 for a discussion of WSU modes.

D-(Field Definition) Specification

The D-specification (Figure 6-1) indicates the position and attributes of data on a display. You must code at least one D-specification after each S-specification. During generation, WSU creates one display screen format from each set of S- and D-specifications. WSU allows up to 245 displays per program. While there can be only one S-specification to describe a display, there can be any number of D-specification lines.

Each D-specification statement can define a variable data field (an input field, output field, or an output/input field), a constant (prompt or constant field value), or both. Refer to the manual, *Creating Displays*, for a more detailed discussion of these fields.

Data Fields

Input Fields

Input fields are fields in which you can enter data. When a program shows a display, input fields are normally blank. The operator then can type data into the input fields; the contents of the field are sent to the program when the operator enters the display. (*Enters the display* refers to the operator pressing the Enter key, pressing a user-enabled command key, or leaving a field for which auto-record-advance has been specified.)

A maximum of 256 fields are allowed in a display. Of those fields, the maximum number of input fields is 127.

Output Fields

Output fields contain information that the operator cannot change, such as the data supplied by the program or constants/prompts. The contents of these fields are not returned to the WSU program when an operator enters the display.

Output/Input Fields

Output/input fields contain information that is either supplied by the program or specified by the display format itself. The operator can change the information in output/input fields. The information is returned to the WSU program when an operator enters the display.

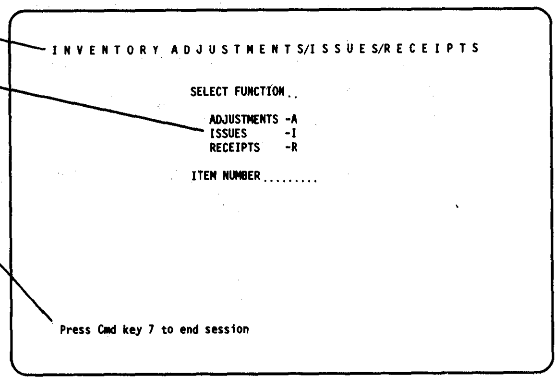
Constant Data

Constant data includes prompts and constant values for fields.

Prompts

Prompts are words on a display that can provide:

- Headings for a display
- Descriptive names for fields
- Operator instructions



```
INVENTORY ADJUSTMENTS/ISSUES/RECEIPTS

      SELECT FUNCTION..

        ADJUSTMENTS -A
        ISSUES      -I
        RECEIPTS    -R

      ITEM NUMBER.........




Press Cmd key 7 to end session
```

Prompts can be specified in quotation marks on the D-specification or stored in your message member and referenced on the D-specification by a 4-digit number. Also, field names from a program's file definition can be used as prompts. Refer to *Constant Type* and *Constant Data* in Chapter 16 on the D-specification for further information on coding prompts.

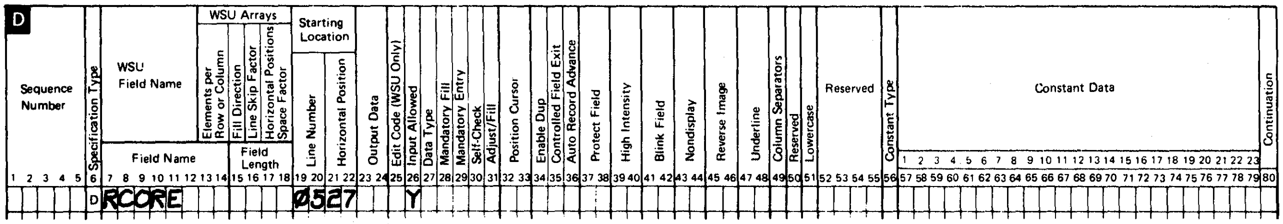
Constant Values

Constant values are first values for input fields that can be specified on the D-specification. For example, a weekly-hours-worked field might be initialized to 40. The constant value is returned to the WSU program when the display is entered. Refer to *Constant Type* and *Constant Data* in Chapter 16 for further information on coding constant values.

Required Entries

You need to identify your field (give its location and, perhaps a name) and specify whether it is for output, input, or both. The name of each of the fields named in your display is identified by a *Field Name* (columns 7 through 12). When you are coding the *Starting Location*, (columns 19 through 22) you need to specify on which line your field will begin and in which position on the line your field will start.

You must also specify whether your field is *Output Data* (columns 23 and 24), that is, only shown on the display or *Input Allowed* (column 26), that is, a field into which data can be entered.



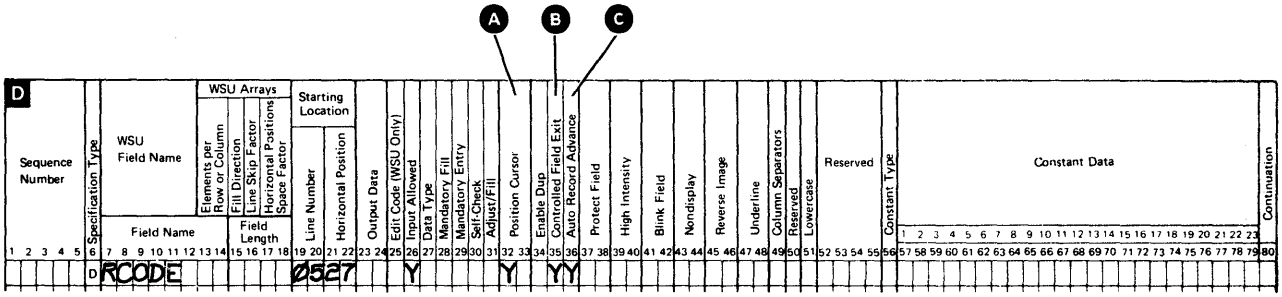
D

Sequence Number 1 2 3 4 5	Specification Type 6	WSU Field Name / Field Name 7 8 9 10 11 12 13 14	WSU Arrays: Elements per Row or Column / Fill Direction / Line Skip Factor / Horizontal Positions Space Factor; Field Length 15 16 17 18	Starting Location: Line Number 19 20	Starting Location: Horizontal Position 21 22	Output Data 23 24	Edit Code (WSU Only) 25	Input Allowed 26	Data Type 27	Mandatory Fill 28	Mandatory Entry 29	Self-Check 30	Adjust/Fill 31	Position Cursor 32 33	Enable Dup 34	Controlled Field Exit 35	Auto Record Advance 36	Protect Field 37 38	High Intensity 39 40	Blink Field 41 42	Nondisplay 43 44	Reverse Image 45 46	Underline 47 48	Column Separators 49	Reserved 50	Lowercase 51	Reserved 52 53 54 55	Constant Type 56	Constant Data 1–23 57–79	Continuation 80
	D	RCORE		05	27			Y																							

Input and Output Fields

When you are defining your fields, you can specify the following for the fields:

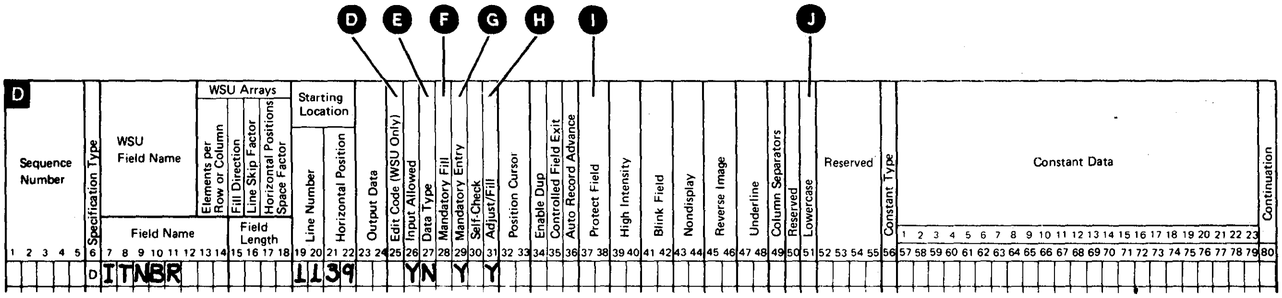
- Movement through fields:
  - *Position Cursor* (columns 32 and 33) allows you to position the cursor on whichever input field you want when a display appears. Ⓐ
  - *Controlled Field Exit* (column 35) specifies whether the cursor advances automatically when a field is filled or if the Field Exit key must be pressed. Ⓑ
  - *Auto Record Advance* (column 36) controls whether all input fields are returned to your program automatically when the last field is entered. Ⓒ



D RCODE 0527 Y Y YY

- What goes in the fields:

  - *Edit Code* (column 25) allows editing (commas, decimal points, etc.) for numeric output fields. Figure 6-4 shows how you can use this entry with the Data/Edit entry in column 43 of the J-specification. Figure 6-5 shows how this entry and the Date/Edit entry in column 43 of the J-specification can be used to edit fields that are from 3 to 6 digits long. For fields that are 7 to 15 digits long, WSU edits the lower order 6 digits and truncates the remaining digits. For example, the unedited field 09111481 would be edited as 11/14/81 if column 25 contained a Y. Ⓓ
  - *Data Type* (column 27) specifies whether the input field is alphabetic, alphameric, numeric, alphameric-katakana, ideographic, or a combination of these types. Ⓔ
  - *Mandatory Fill* (column 28) specifies that the operator must enter all or none of the input. Ⓕ
  - *Mandatory Entry* (column 29) specifies that the operator must enter at least one character or blank in the field. Ⓖ
  - *Adjust/Fill* (column 31) specifies if the input field is to be right adjusted and filled with zeros or blanks. Ⓗ
  - *Protect Field* (columns 37 and 38) specifies if the field will be skipped or not. Ⓘ
  - *Lowercase* (column 51) specifies that all alphabetic characters typed from the the keyboard can be displayed and sent to the user program in lowercase (if the Shift key is not used) or uppercase (if the Shift key is used). Ⓙ



Date/Edit Entry (Column 43 J-Specification)	3-Digit Negative Field No Decimal Positions		5-Digit Negative Field 2 Decimal Positions		7-Digit Negative Field 2 Decimal Positions	
	Unedited	Edited	Unedited	Edited	Unedited	Edited
M	12 N	125	12345	123.45	123454N	12,345.45
D	12 N	125	12345	123,45	123454N	12.345,45
Y	12 N	125	12345	123,45	123454N	12.345,45

Notes:

1. Edited refers to an edit code of J in column 25 of the D-specification. Unedited refers to a blank edit code.
2. If a field is an output/input field and its data type is S, an unedited negative value appears with the minus sign as shown in the edited examples.

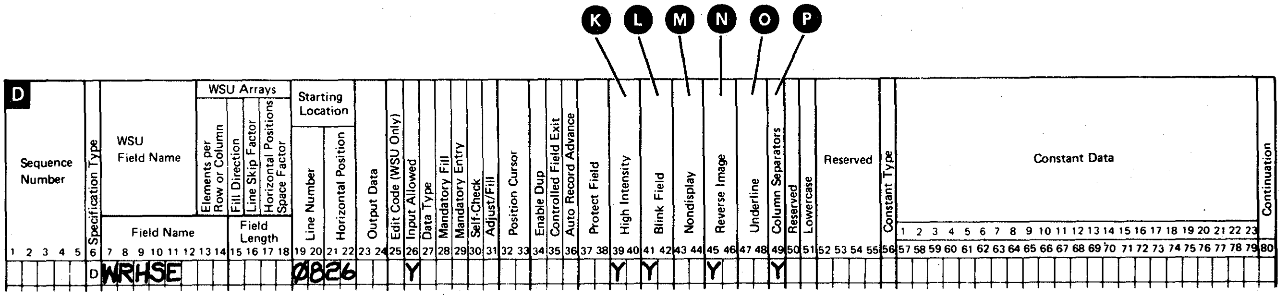
Figure 6-4. Examples of a J Edit Code Used to Edit Numeric Output Fields

Date/Edit Entry (Column 43 J-Specification)	3-Digit Field		4-Digit Field		5-Digit Field		6-Digit Field	
	Unedited	Edited	Unedited	Edited	Unedited	Edited	Unedited	Edited
M	119	11/9	1179	11/79	11149	11/14/9	111479	11/14/79
D	119	11/9	1179	11/79	14119	14/11/9	141179	14/11/79
Y	911	9/11	7911	79/11	91114	9/11/14	791114	79/11/14

Note: Edited refers to an edit code of U in column 25 of the D-specification. Unedited refers to a blank edit code.

Figure 6-5. Example of a Y Edit Code Used to Edit Numeric Output Fields

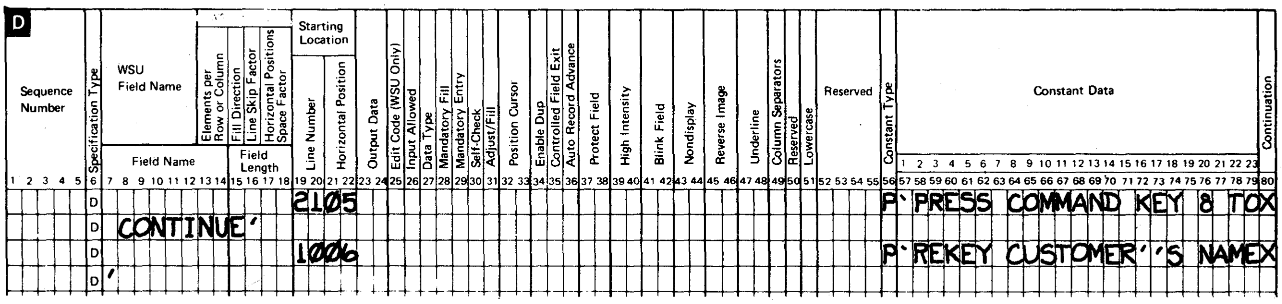
- How the field will look:
  - *High Intensity* (columns 39 and 40) specifies that the field be brighter than normal. Ⓚ
  - *Blink Field* (columns 41 and 42) specifies that the field blink. Ⓛ
  - *Nondisplay* (columns 43 and 44) specifies whether the data is displayed or not. Ⓜ
  - *Reverse Image* (columns 45 and 46) specifies that the data image be reversed (dark print on light background instead of the usual light print on a dark background). Ⓝ
  - *Underline* (columns 47 and 48) specifies that the data on the display be underlined. Ⓞ
  - *Column Separator* (column 49) specifies that column separators be inserted between the characters on a field. Ⓟ



```
     DWRHSE       0826   Y            Y Y   Y   Y
```

Constants

Use *Constant Type* (column 56) and *Constant Data* (columns 57 through 79) when you want WSU to supply a field when your display appears. If the constant is longer than the *Constant Data* field allows, use the *Continuation* (column 80) to code any nonblank character to show that your constant overflows to the following line. See Figure 6-6 for an example of continuing constant data. Figure 6-7 shows the D-specification entries for coding fields and constant data. Figure 6-8 shows more examples of coding fields, prompts, and constant data.



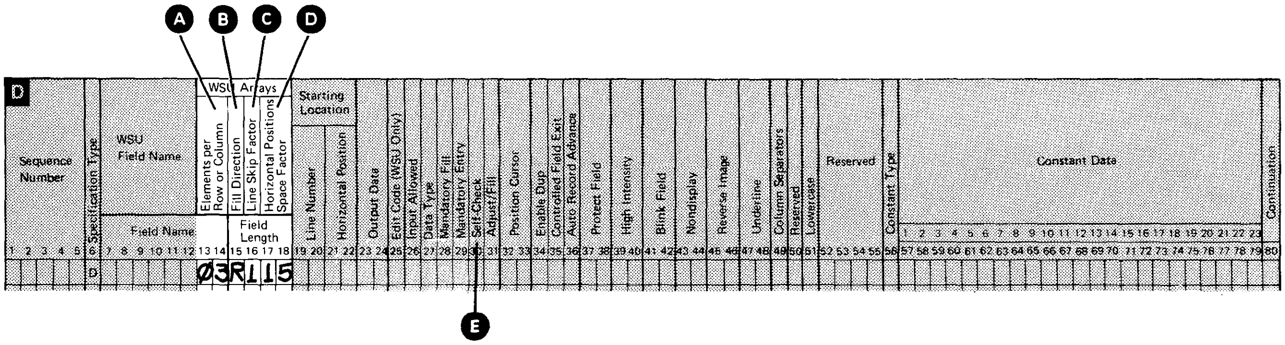
```
     D            2105                                 P`PRESS COMMAND KEY 8 TOX
     DCONTINUE'
     D            1006                                 P`REKEY CUSTOMER''S NAMEX
     D'
```

Figure 6-6. Examples of Continued Constant Data

WSU Arrays

The following information is used only when you are entering or displaying execution-time arrays from the display. Arrays also require coding the E-specification:

- *Elements Per Row or Column* (columns 13 and 14) specifies the number of elements included per row or per column in your array. Ⓐ
- *Fill Direction* (column 15) specifies whether the array elements appear vertically in columns or horizontally in rows. Ⓑ
- *Line Skip Factor* (column 16) specifies the number of lines to leave between the rows of elements in your array. Ⓒ
- *Horizontal Positions Space Factor* (columns 17 and 18) specifies the number of spaces to leave between the columns of elements in your array. Ⓓ



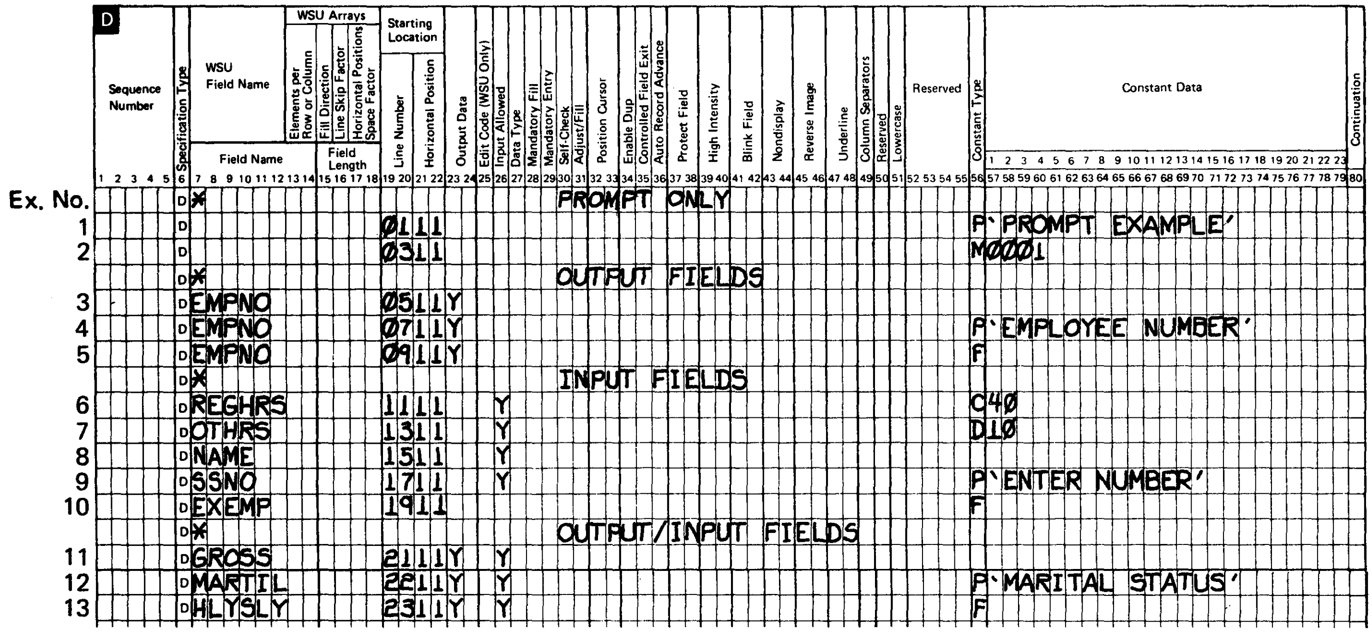
Self-Check

Use *Self-Check* Ⓔ to get limited protection against clerical or typing errors. The explanation and formula for using Self-Check is covered in Appendix A, *Using Self-Check Fields.*

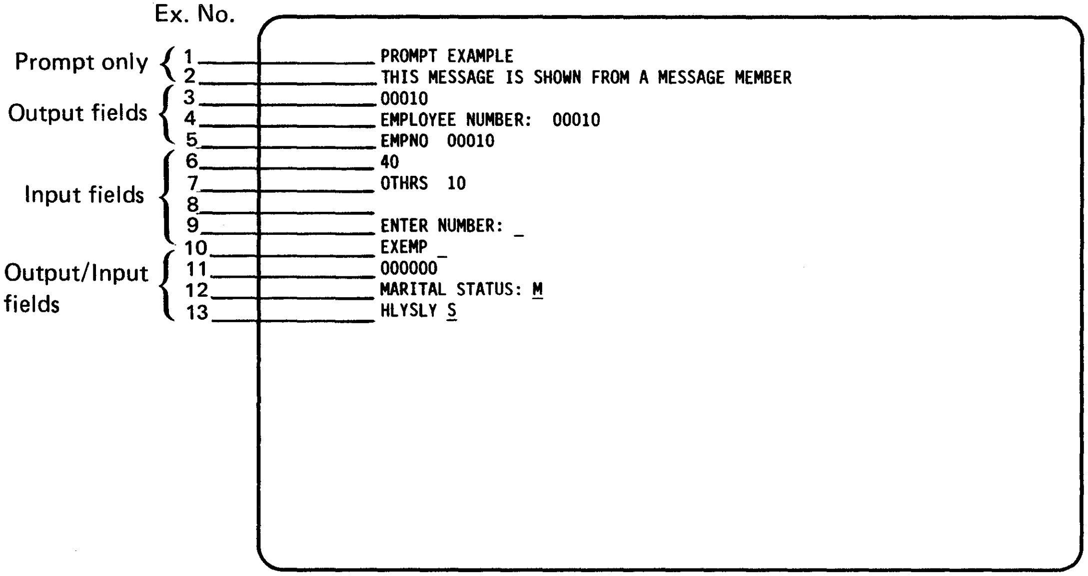
				Sequence Number 1 2 3 4 5	Specification Type 6	WSU Field Name Field Name 7 8 9 10 11 12 13 14	WSU Arrays Field Length 15 16 17 18	Starting Location Line Number 19 20	Starting Location Horizontal Position 21 22	Output Data 23 24	Edit Code (WSU Only) 25	Input Allowed 26	Data Type 27	Mandatory Fill 28	Mandatory Entry 29	Self-Check 30	Adjust/Fill 31	Position Cursor 32 33	Enable Dup 34	Controlled Field Exit 35	Auto Record Advance 36	Protect Field 37 38	High Intensity 39 40	Blink Field 41 42	Nondisplay 43 44	Reverse Image 45 46	Underline 47 48	Column Separators 49	Reserved 50	Lowercase 51	Reserved 52 53 54 55	Constant Type 56	Constant Data 57–80
1	Prompt Only	Via prompt in quotes			D			RR	RR														OO	OO	OO	OO	OO	O				P	\`Prompt'
2		Via MIC			D			RR	RR														OO	OO	OO	OO	OO	O				M	NNnn
					D																												
3	Output Field	Without a prompt			D	RRRRRR		RR	RR	RR	O											OO	OO	OO	OO	OO	OO	O					
4		With a descriptive prompt			D	RRRRRR		RR	RR	RR	O											OO	OO	OO	OO	OO	OO	O				P	\`Prompt'
5		With a field name prompt			D	RRRRRR		RR	RR	RR	O											OO	OO	OO	OO	OO	OO	O				F	
					D																												
6	Input Field	With a constant initial value	Without a prompt		D	RRRRRR		RR	RR			R	O	O	O	O	O	OO		O	O		OO	OO	OO	OO	OO	O				C	CONSTANT
7			With a field name prompt		D	RRRRRR		RR	RR			R	O	O	O	O	O	OO		O	O		OO	OO	OO	OO	OO	O				D	CONSTANT
8		No initial value	Without a prompt		D	RRRRRR		RR	RR			R	O	O	O	O	O	OO		O	O		OO	OO	OO	OO	OO	O					
9			With a descriptive prompt		D	RRRRRR		RR	RR			R	O	O	O	O	O	OO		O	O		OO	OO	OO	OO	OO	O				P	\`Prompt'
10			With a field name prompt		D	RRRRRR		RR	RR			R	O	O	O	O	O	OO		O	O		OO	OO	OO	OO	OO	O				F	
					D																												
11	Output/Input Field	Without a prompt			D	RRRRRR		RR	RR	RR		R	O	O	O	O	O	OO		O	O	OO	OO	OO	OO	OO	OO	O					
12		With a descriptive prompt			D	RRRRRR		RR	RR	RR		R	O	O	O	O	O	OO		O	O	OO	OO	OO	OO	OO	OO	O				P	\`Prompt'
13		With a field name prompt			D	RRRRRR		RR	RR	RR		R	O	O	O	O	O	OO		O	O	OO	OO	OO	OO	OO	OO	O				F	

Legend: R — Required entries (in addition to those entries in columns 56 through 80)
O — Optional entries

Figure 6-7. Coding Fields and Constant Data on D-Specifications



```
     D*                      PROMPT ONLY
     D            0111                                 P'PROMPT EXAMPLE'
     D            0311                                 M0001
     D*                      OUTPUT FIELDS
     DEMPNO       0511Y
     DEMPNO       0711Y                                P'EMPLOYEE NUMBER'
     DEMPNO       0911Y                                F
     D*                      INPUT FIELDS
     DREGHRS      1111   Y                             C40
     DOTHRS       1311   Y                             D10
     DNAME        1511   Y
     DSSNO        1711   Y                             P'ENTER NUMBER'
     DEXEMP       1911                                 F
     D*                      OUTPUT/INPUT FIELDS
     DGROSS       2111Y  Y
     DMARTIL      2211Y  Y                             P'MARITAL STATUS'
     DHLYSLY      2311Y  Y                             F
```



```
PROMPT EXAMPLE
THIS MESSAGE IS SHOWN FROM A MESSAGE MEMBER
00010
EMPLOYEE NUMBER:  00010
EMPNO  00010
40
OTHRS  10

ENTER NUMBER: _
EXEMP _
000000
MARITAL STATUS: M
HLYSLY S
```

Figure 6-8. Examples of Coding Fields, Prompts, and Constant Data

Chapter 7. Coding Processing

C-(processing) specifications provide the instructions for directing data to a display station, a transaction file, or a master file, as well as for processing data (add, subtract, and so forth). Figure 7-1 shows the C-specification coding form. The number of C-specifications following a set of S- and D-specifications depends upon the type of processing that needs to be done. There may be times that you do not need C-specifications for a display, but for most of your programs, you will.

This chapter explains:

- The entries that you can code on the C-specification and provides coding examples of these entries. Additional information about coding C-specifications can be found in:
  - Chapter 2, which describes the WSU program cycle and processing levels.
  - Chapter 3, which has more detailed information about the indicators you can use on the C-specification.
  - Chapter 11, which contains example programs for which C-specifications have been coded.
  - Chapter 17, which provides details of the C-specification entries.
  - Chapter 18, which contains detailed explanations of the operation codes.
- The subroutines used on your C-specifications.
- The preprocessing that can be specified on your C-specification before the display appears.
- The operations that can be done using C-specifications.

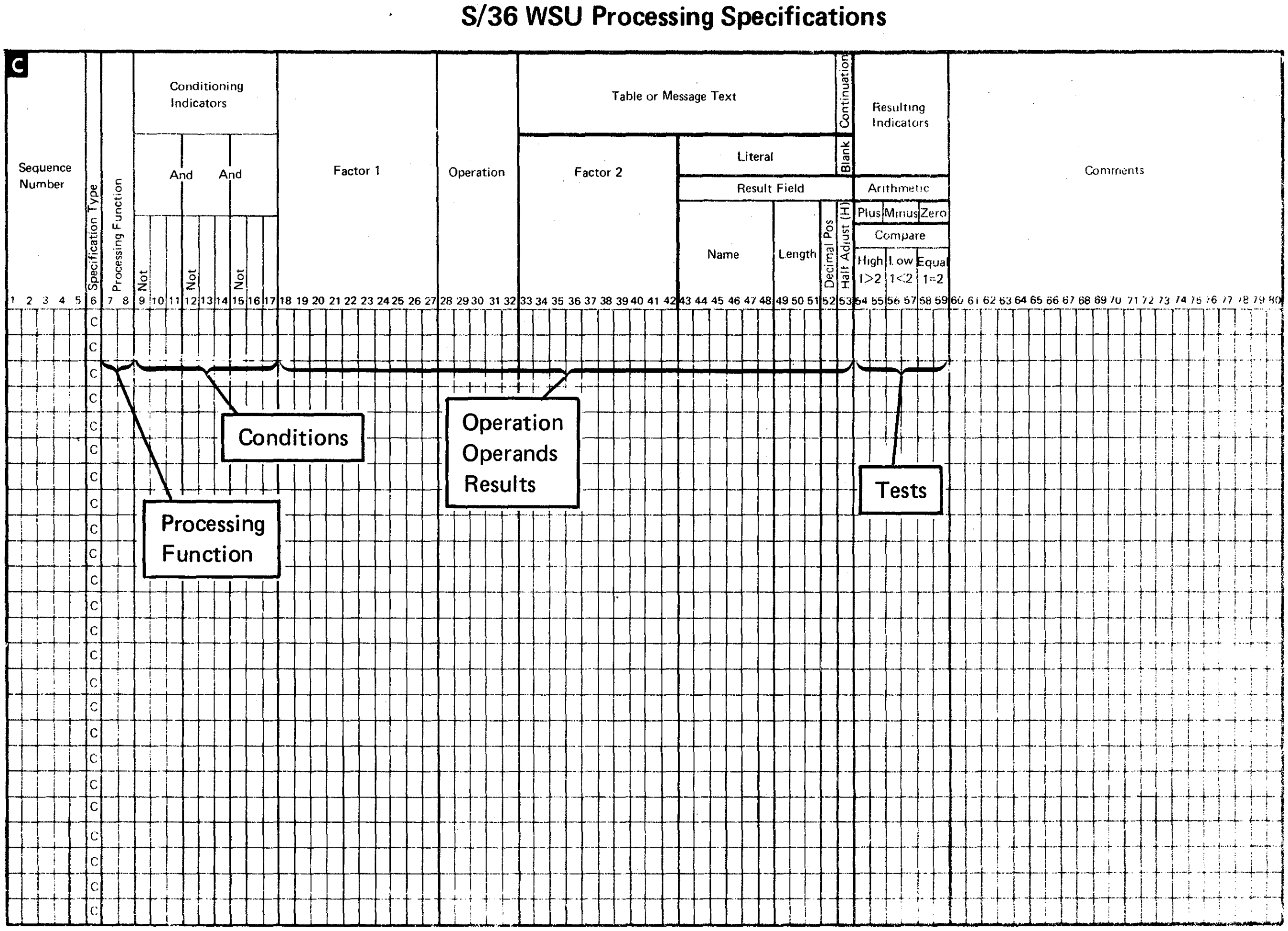


Figure 7-1. C-Specification Coding Form

C-Specification Entries

Each C-specification has four areas:

- A specified processing function
- Conditions under which operations occur
- Operations, operands, and results
- Tests made on the results of the operations, and resulting indicators that reflect the results of the tests.

Note: A given instruction that you code on the C-specification may not include an entry for each of the above areas.

Processing Function

You can set the *Processing Function* (columns 7 and 8) to specify indicators that WSU turns on automatically when the processing function begins, or that you set on to cause any processing for that function to occur. The processing function indicators used by WSU are:

IJ	(job-initiation processing level)
IW	(work-session processing level)
ES	(end-of-sequence-set processing level)
EW	(end-of-work-session processing level)
SR	(subroutine operations)
AN or OR	(to use a combination of the processing level indicators)
EJ	(end-of-job processing level)

Conditions

Use *Conditioning Indicators* (columns 9 through 17) to specify the conditions for doing the operations you want done for your program. You can code more than one indicator for your program by using AN or OR. For more information on indicators see Chapter 3, *Reserved Fields and Indicators*. The conditioning indicators used by WSU are:

Blank	(normal enter mode)*
AC	(accept command key)
AE	(accept sequence error)
CG	(current group)
DL	(delete record)
EJ	(end-of-job processing level)*
ES	(end-of-sequence-set processing level)*
EW	(end-of-work-session processing level)*
IJ	(job-initiation processing level)*
IN	(insert mode)
IP	(input to process)
IS	(initiate transaction sequence)
IW	(work-session-initiation processing level)*
JA-JN, JP-JY	(job)
KG-KL, KQ-KY	(command key)

PG	(program mode)
RC	(recovery of work session)
RP	(repeated display)
RR	(review record)
RS	(resume work session)
RU	(rover update)
RV	(review mode)
SA-SN, SP-SY	(session level)
U1-U8	(external)
01-89	(general indicator)

Note: An * after the indicator description means that these indicators can also be used in *Processing Function*.

Operation Codes

Factor 1 (columns 18 through 27) and *Factor 2* (columns 33 through 42) name the fields (constants) or specify the data (literals) on which you want to perform your operations. See Chapter 17 for a discussion of literals. Refer to *Operations* in this chapter for more about the operations you can perform.

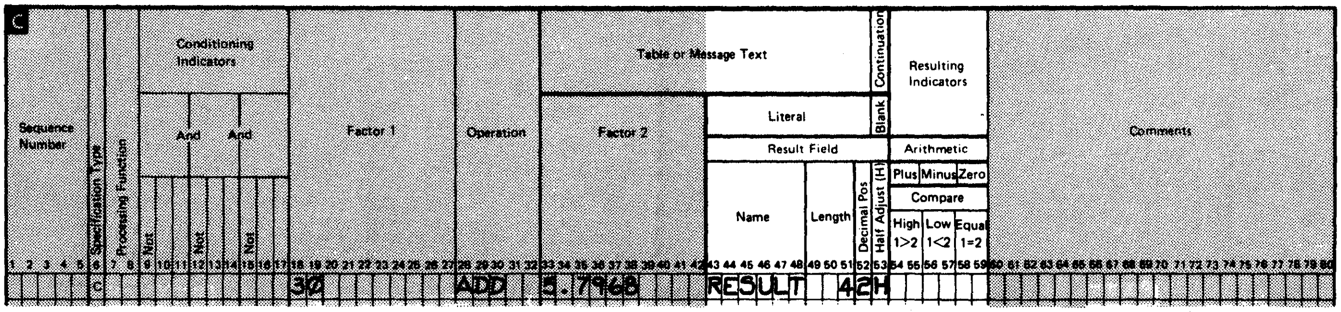
Operation codes can be grouped in the following categories:

- Arithmetic:
  - Add (ADD)
  - Subtract (SUB)
  - Multiply (MULT)
  - Divide (DIV)
  - Move remainder (MVR)
  - Zero and add (Z-ADD)
  - Zero and subtract (Z-SUB).
- Arrays
  - Look for specific element of array (LOKUP)
  - Move data between fields (MOVEA)
  - Sorting elements (SORTA)
  - Summing the values of an array (XFOOT).
- Branching (within C-specifications)
  - Go to (GOTO)
  - Tag (TAG)
  - Execute subroutine (EXSR).

- Comparing and testing
    - Compare (COMP)
    - Range check (RANGE).
- Debugging (DEBUG)
- Indicators (setting on and off)
    - Set on (SETON)
    - Set off (SETOF).
- Input and output
    - Master files
        - Get record (GET)
        - Write record (PUT)
        - Add new record (PUTN).
    - Transaction file
        - Get next header (GETNH)
        - Get next record (GETNR)
        - Get previous header (GETPH)
        - Get previous record (GETPR)
        - Write record (PUT).
    - Messages/Displays
        - Display information message (IMSG)
        - Display diagnostic message (MSG)
        - Show a display (PUTS).
- Move
    - Move (MOVE)
    - Move left (MOVEL).
- Subroutine
    - Begin subroutine (BEGSR)
    - Assign priority (PRTY)
    - End subroutine (ENDSR).

Results

Result Field (columns 43 through 52) includes descriptions of the results of the operations performed. *Name* (columns 43 through 48) and *Length* (columns 49 through 51) name the field and the length of the field that contains the results of the operation or the fields or literal upon which the operation is done. The result field can also be broken down into the number of *Decimal Positions* (column 52) and *Half Adjust* (rounded) (column 53).



Specification Type: C
Factor 1: 30
Operation: ADD
Factor 2: 5.7968
Result Field Name: RESULT
Length: 4
Decimal Pos: 2
Half Adjust (H): H

Code the field length for a result field long enough to hold the longest possible result. If the result field is too short, significant digits can be lost, or the result can be unpredictable depending on the contents of the fields at the time the operation is performed. For example, to add field A (8 positions long with 4 decimal places) to field B (10 positions long with 6 decimal places), field C (the result field) must allow at least 5 positions to the left and 6 positions to the right of the decimal.

9999.0000	Field A
0001.111111	Field B
10000.111111	Field C (result field)

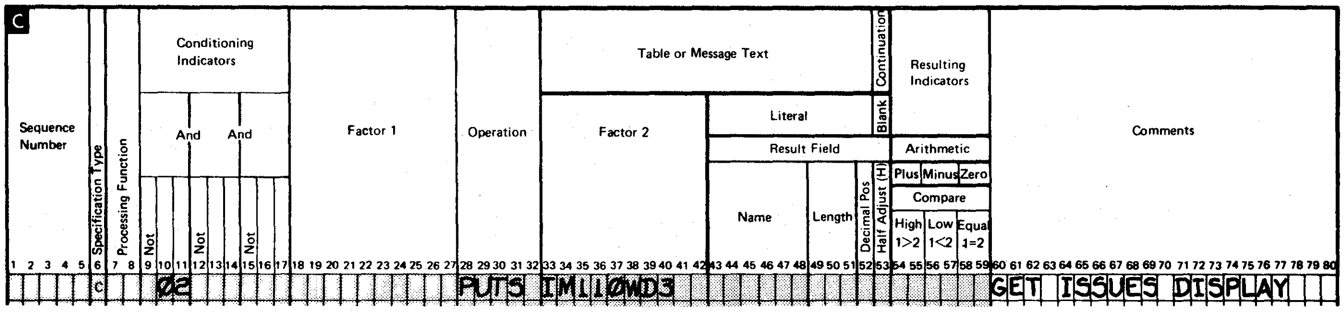
In this case, field C was defined as 11 positions long with 6 decimal places. Some of the numbers to the right of the decimal could be lost without significantly changing the meaning of the result. However, if field C was defined as 10 positions long with 6 decimal positions, a significant digit to the left of the decimal would be lost. Field C in this case would be 0000.111111; the meaning of the result has significantly changed.

Those result fields defined as packed fields on I-specifications can have their lengths defined again by this entry. The result field length must allow for the unpacked length of a field.

For example, assume FIELD3 is a packed field, 4 positions long. If FIELD3 is used as a result field on a C-specification, the length can be specified either as 6 or as 7. If a length is not specified, WSU assumes the greater of the two possibilities, which in this example is 7. *Resulting Indicators* (columns 54 through 59) indicate the results of an arithmetic operation or specify which indicators are to be set on or off for your operations.

Comments

While *Comments* (columns 60 through 80) do not affect WSU operation, they do help document your program.



Sequence Number (1-5)	Specification Type (6)	Processing Function (7-8)	Conditioning Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-53)	Resulting Indicators (54-59)	Comments (60-80)
	C		02		PUTS	IM110WD3			GET ISSUES DISPLAY

Subroutines

A subroutine consists of one or more operations that you code at the end of your program. Subroutines consist of C-specifications only; they do not have associated displays.

In the following description of subroutines, letters Ⓐ through Ⓔ reference items in Figure 7-2.

Begin a subroutine by coding its name in factor 1 followed by the BEGSR operation code Ⓑ. A subroutine name can be 1 to 6 characters long, and must begin in column 18 with an alphabetic character. Remaining characters must be alphameric with no embedded blanks. Each subroutine in your program must have a name. A subroutine cannot have the same name as another subroutine, a label for a TAG or ENDSR operation, or a field defined in the program.

If a subroutine requires priority, code the PRTY operation after the BEGSR operation. Refer to Chapter 18, *Operation Codes*, for an explanation of the PRTY operation.

An entry of SR in Processing Function (columns 7 and 8) is optional on each subroutine C-specification Ⓓ. Indicate the last line of a subroutine with the ENDSR operation Ⓒ. You must code ENDSR before you can code another subroutine.

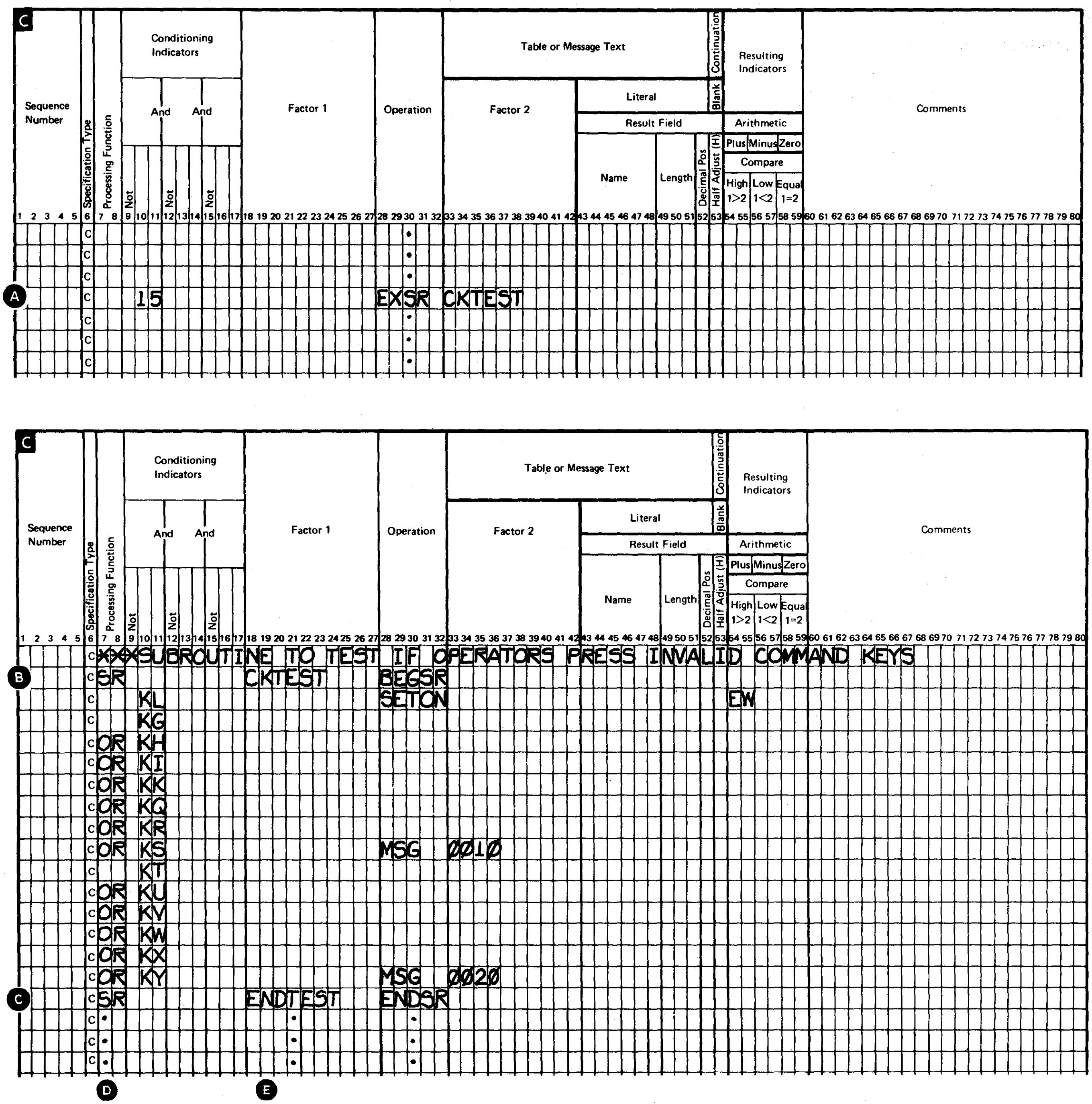
Factor 1 of the ENDSR operation can contain a name Ⓔ. This name indicates a line to which a GOTO within the subroutine can branch.

The GOTO operation within a subroutine can branch only to another line in the same subroutine. You can condition operations within a subroutine by indicators in columns 9 through 17. Also, fields used by the subroutine can be defined within or outside of the subroutine.

If the MSG operation is used in a subroutine, the current display is shown, and processing returns to the first C-specification for the display.

You can code several subroutines in your program in any order; however, you cannot code a subroutine within a subroutine. You can code all operations except EXSR in a subroutine. This means that a subroutine cannot call another subroutine or itself.

Even though you code subroutines at the end of your program, they are run whenever an EXSR operation calls the subroutine Ⓐ (Figure 7-2). EXSR causes a branch to the subroutine named in factor 2. When the subroutine ends, the operation following the EXSR operation receives control. You can use indicators in columns 9 through 17 to condition when the EXSR operation should run.



```
     C    15                EXSR CKTEST
```

```
     C***SUBROUTINE TO TEST IF OPERATORS PRESS INVALID COMMAND KEYS
     CSR         CKTEST    BEGSR
     C    KL                SETON                     EW
     C    KG
     COR  KH
     COR  KI
     COR  KK
     COR  KQ
     COR  KR
     COR  KS                MSG  0010
     C    KT
     COR  KU
     COR  KV
     COR  KW
     COR  KX
     COR  KY                MSG  0020
     CSR         ENDTEST   ENDSR
```

Figure 7-2. Sample Subroutine

Preprocessing

Preprocessing allows you additional control of displays.

You can use preprocessing to:

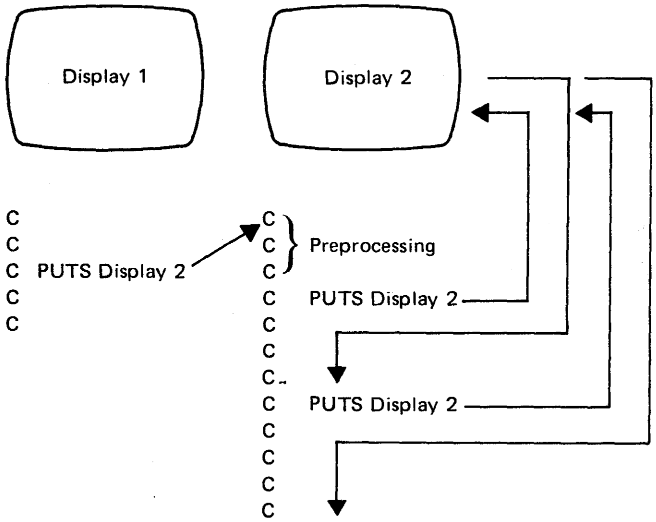
- Initialize values of output fields or output/input fields.
- Determine whether to show a display.
- Back out totals during review mode.
- Provide a common end-of-processing routine for several displays.

WSU does not automatically show a display that has Y coded in *Preprocess* (column 47) of its S-specification. Instead, WSU first performs the C-specifications for the display. Then to show the display, you can issue a PUTS operation for the display. If a PUTS operation causes the display to appear, operators can respond to the display. When an operator enters the display, processing resumes with the operation following the PUTS operation.

You can code any number of PUTS operations to the preprocessed display.

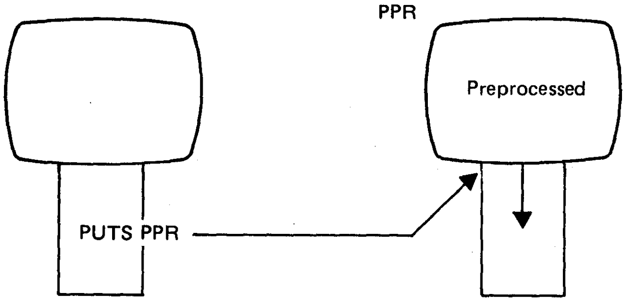
When a PUTS occurs for a preprocessed display from another display (display 1 uses a PUTS to show display 2 in the following example), processing starts at the first C-specification, and the display is not shown.

A display can issue a PUTS operation for itself. After the display is shown, processing resumes at the operation that follows the PUTS.

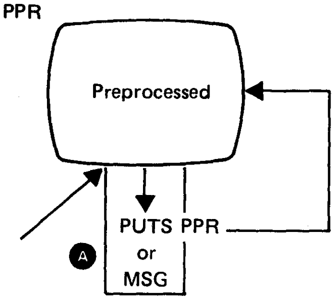


You must code C-specifications for displays that WSU preprocesses. Because a display that does not allow operator input cannot have associated processing, do not specify N for Reset Keyboard (column 23 of an S-specification) and preprocessing for the same display.

As the following shows, a PUTS to a preprocessed display from another display causes the processing to begin.



A PUTS or MSG operation within the preprocessed display causes the display to appear Ⓐ. An IMSG operation does not cause the display to appear; each IMSG operation only adds a message to the bottom line of the current display.



If an operator selects a preprocessed display by means of the WSU display, or if a preprocessed display is selected by WSU for review mode or insert mode, C-specification processing begins, and a PUTS operation is required to show the display.

Operations

WSU allows you to perform many different types of operations on your data. Special codes, which are entered on the processing specifications, indicate the operation to be performed. Usually these codes are just abbreviations of the name of the operation.

Arithmetic Operations

The arithmetic operations are:

- ADD adds factor 1 to factor 2 and places the sum in the result field.
- DIV (divide) divides factor 1 (dividend) by factor 2 (divisor) and places the result (quotient) in the result field.
- MULT (multiply) multiplies factor 1 by factor 2 and places the product in the result field.
- MVR (move remainder): moves the remainder from the previous divide operation to the result field.
- SUB (subtract) subtracts factor 2 from factor 1 and places the difference in the result field.
- Z-ADD (zero and add) places zeros in the result field and then adds factor 2 to the field and places the sum in the result field.
- Z-SUB (zero and subtract) places zeros in the result field and then subtracts factor 2 from the field and places the difference in the result field.

For these operations, factor 1 and factor 2 (if not blank) must be numeric fields or numeric literals, and the result field must be numeric.

All arithmetic operations do decimal alignment. Even though truncating may occur, the position of the decimal point in the result field does not change. For arithmetic operations that use all three fields, factor 1, factor 2, and the result field, two or all can be the same field, or all can be different fields.

The length of any field in an arithmetic operation cannot exceed 15 digits. If the result exceeds 15 digits, digits may drop from either or both ends depending on the location of the decimal point. The results of all operations are signed (+ or -), and the sign is in the zoned portion of the rightmost digit. Refer to Chapter 4, *Coding Files* for a discussion of data formats.

Arithmetic operations cannot result in negative zero. A zero has a plus zone. You can code the following indicators in columns 54 through 59 to signal whether the results of an arithmetic operation are plus (columns 54 and 55), minus (columns 56 and 57), or zero (columns 58 and 59):

AE

EJ

ES

EW

JA-JN, JP-JY

KG-KL, KQ-KY

SA-SN, SP-SY

U1-U8

01-89

Refer to Chapter 3, *Reserved Fields and Indicators,* for an explanation of these indicators.

Array Operations

WSU provides the following operations for array operations in a WSU program:

- LOKUP (look for specific element of array) causes the search to be made for the next instance of a specific element in an array. If the searched-for element is found, the number or position of this element (not its contents) is placed in the index field. In this way, the actual element that satisfies the LOKUP operation can be used in later calculations.

  For examples of looking up arrays, see Figures 7-3 and 7-4.

- MOVEA (move array data between arrays and array fields) transfers characters from the leftmost positions of factor 2 to the leftmost positions of the result field.

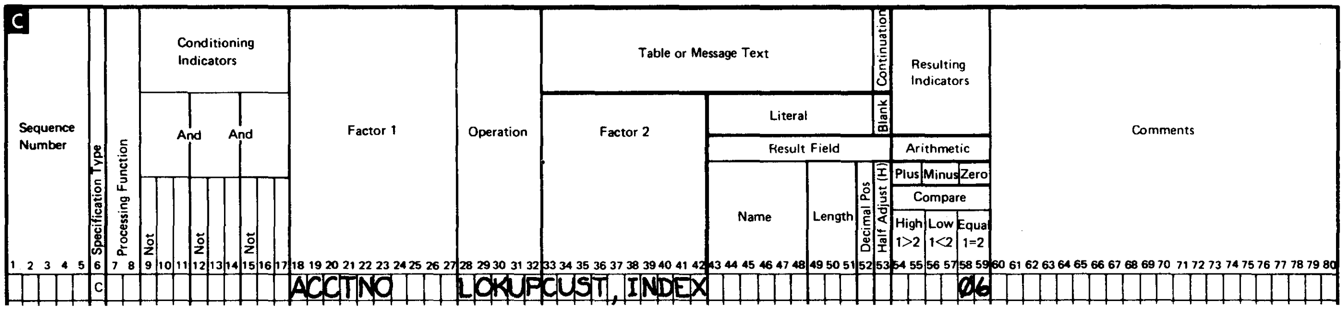
  Figure 7-5 illustrates the use of the MOVEA operation.

- SORTA (sorting elements) allows you to sequence the elements of an array during running of a program. You can ensure that the elements of the array are in the proper sequence for a LOKUP operation by performing a SORTA operation. If you have an array of ideographic characters, do not use SORTA to sequence the array.

  For examples of the SORTA operation, see Figure 7-6.

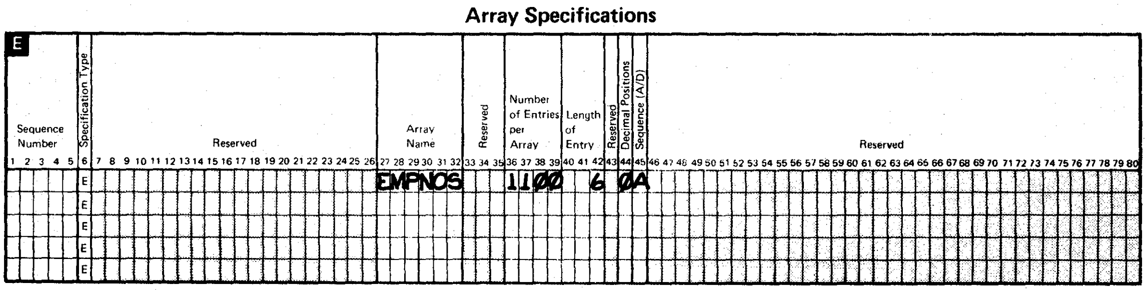
  XFOOT (summing the values of an array) is used only on numeric arrays to add the elements of the array together and place the sum into the field specified as the result field.

For information on how to code arrays and array operations, refer to Chapter 5, *Coding Arrays*, and Chapter 14, *E-(Array) Specifications*.



Sequence Number (1-5)	Specification Type (6)	Processing Function (7-8)	Conditioning Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Pos (52)	Half Adjust (H) (53)	Resulting Indicators High 1>2 (54-55)	Resulting Indicators Low 1<2 (56-57)	Resulting Indicators Equal 1=2 (58-59)	Comments (60-80)
	C			ACCTNO	LOKUP	CUST,INDEX							06	

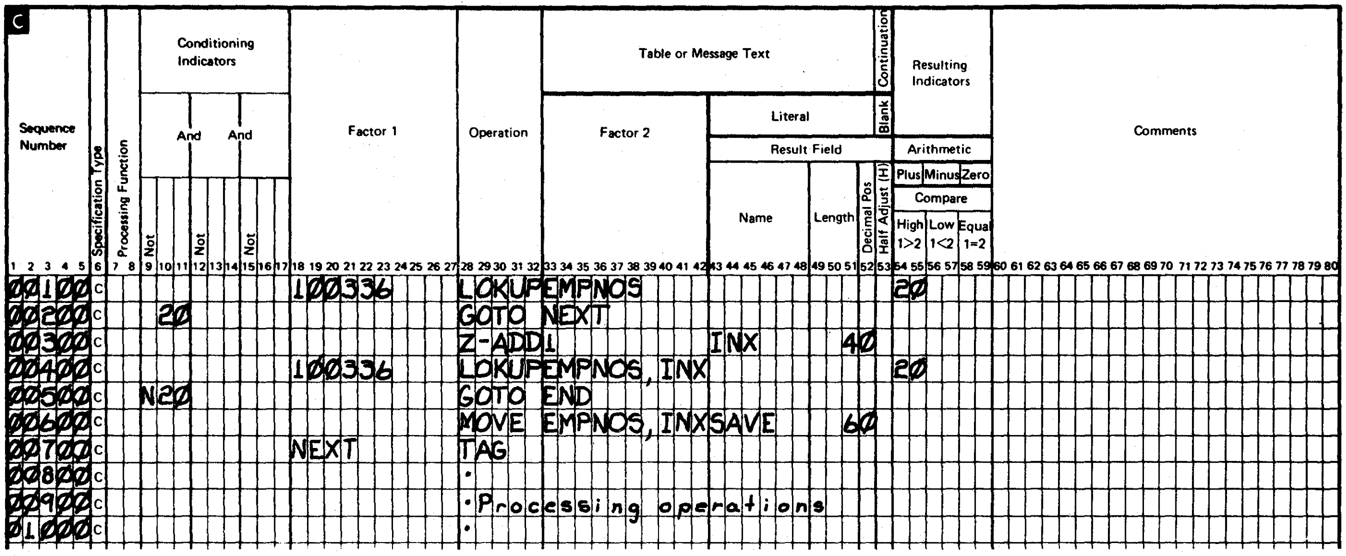
Figure 7-3. Array Lookup: Starting at a Particular Array Item



```
     E                    EMPNOS   1100  6 0A
```

Figure 7-4. LOKUP Operation for an Array

EMPNOS, a 1100-element array of employee numbers, is read in at execution time from the file.



```
00100C           100336    LOKUPEMPNOS               20
00200C   20                GOTO NEXT
00300C                     Z-ADD1         INX     40
00400C           100336    LOKUPEMPNOS,INX           20
00500C  N20                GOTO END
00600C                     MOVE EMPNOS,INXSAVE    60
00700C           NEXT      TAG
00800C                     .
00900C                     .Processing operations
01000C                     .
```

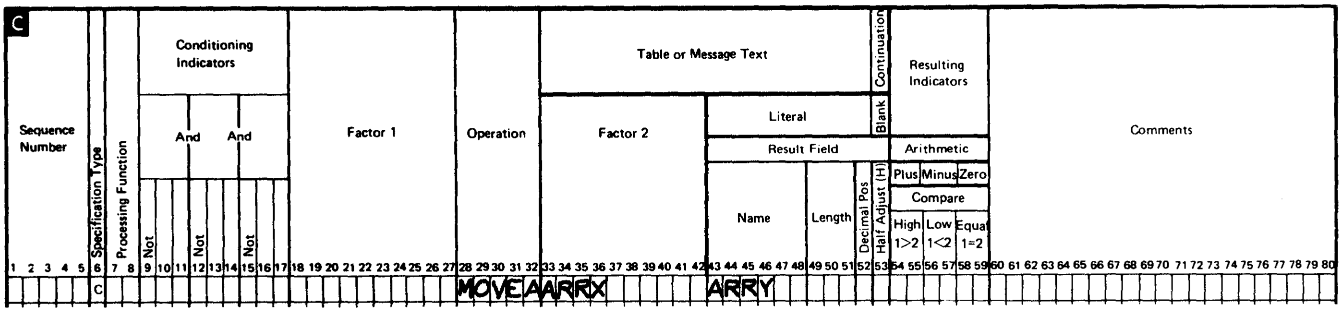
Sequence Numbers 00100 and 00200

The first calculation specification is a LOKUP of EMPNOS to find the element nearest to, but higher in sequence than, the search word 100336. If this element is found in the array, indicator 20 turns on, and the GOTO in line 00200 is performed. Indicator 20 indicates only whether or not the searched-for element exists in the array.

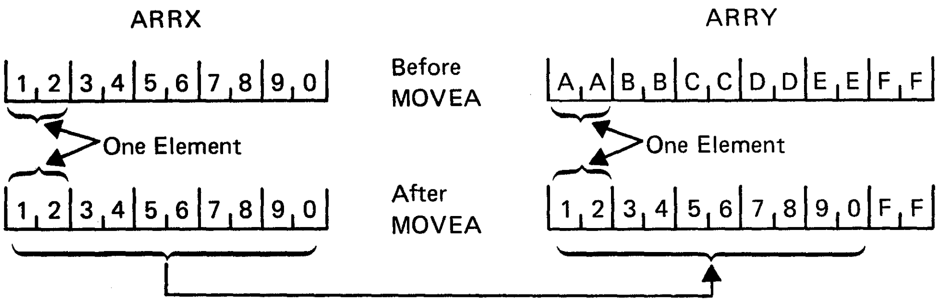
Sequence Numbers 00300 through 01000

The specification on line 00400 shows essentially the same LOKUP operation. Indicator 20 turns on when the first element higher in sequence than 100336 is found. However, in this LOKUP operation, the array EMPNOS is indexed by the field INX. This index field was set to 1 in line 00300, so the LOKUP begins at the first element of EMPNOS. If the searched-for element is found, the number of this element (not its contents) is placed in the field INX. In this way, the actual element that satisfied the LOKUP can be used in subsequent calculation operations, as in line 00600. If no element was found to satisfy the LOKUP, the field INX is reset to 1.

S/36 WSU Processing Specifications

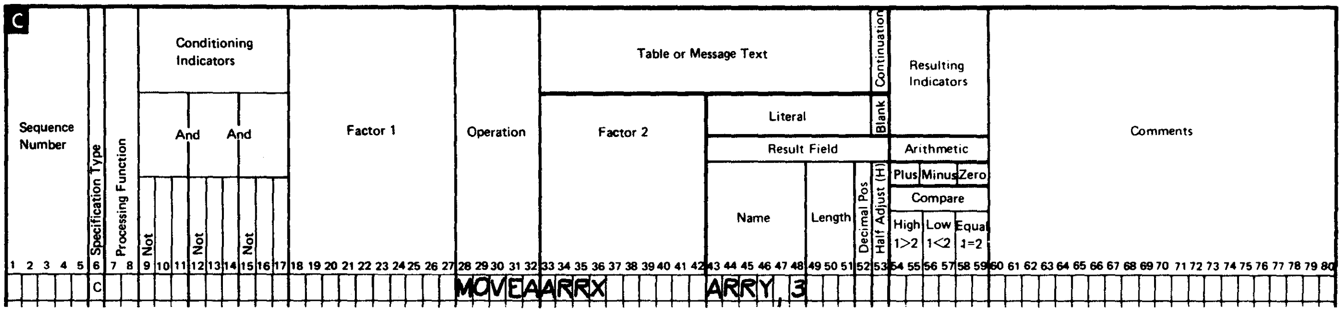


Sequence Number (1-5)	Specification Type (6)	Processing Function (7-8)	Conditioning Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Pos (52)	Half Adjust (H) (53)	Resulting Indicators (54-59)	Comments (60-80)
	C				MOVEA	ARRX	ARRY					



Example: Array-to-array move.
Index result field.

S/36 WSU Processing Specifications



Sequence Number (1-5)	Specification Type (6)	Processing Function (7-8)	Conditioning Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Pos (52)	Half Adjust (H) (53)	Resulting Indicators (54-59)	Comments (60-80)
	C				MOVEA	ARRX	ARRY,3					

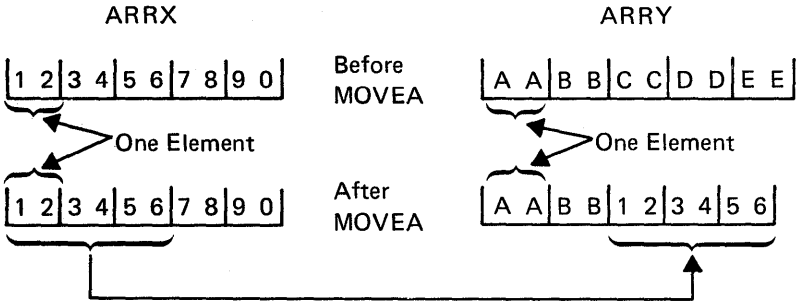
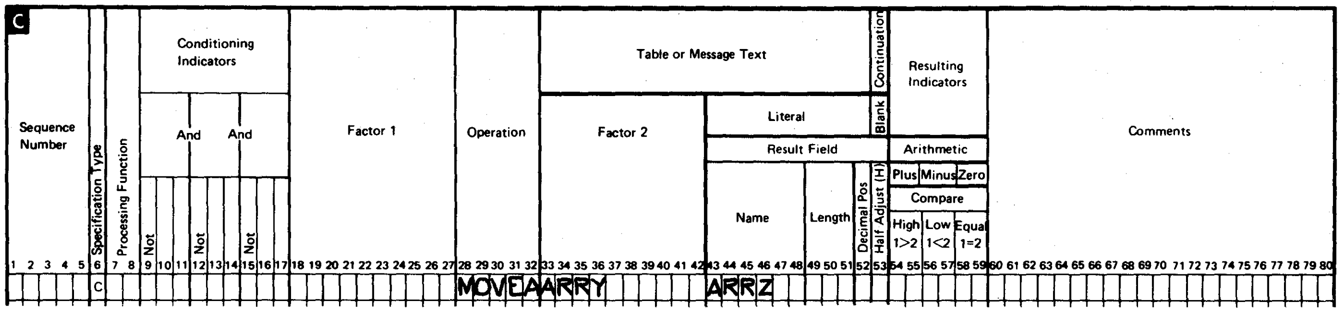


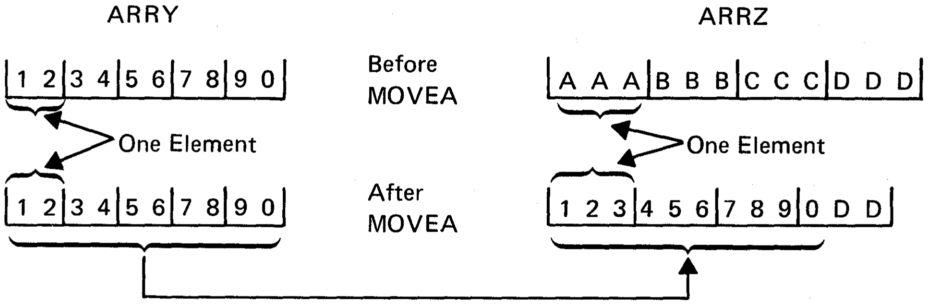
Figure 7-5 (Part 1 of 3). MOVEA Operation

Example: Array-to-array move.
No indexing, different length
array elements.

S/36 WSU Processing Specifications

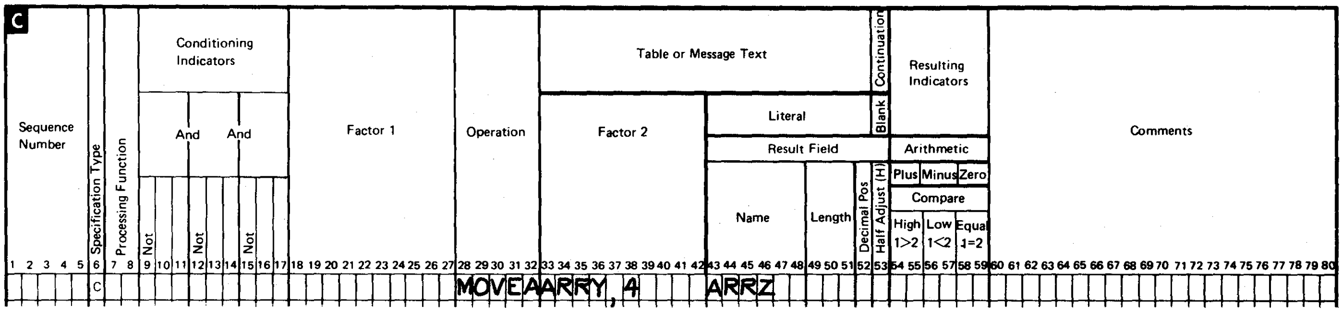


Specification Type	Operation	Factor 2	Result Field Name
C	MOVEA	ARRY	ARRZ



Example: Array-to-array move.
Index factor 2, different length
array elements.

S/36 WSU Processing Specifications



Specification Type	Operation	Factor 2	Result Field Name
C	MOVEA	ARRY,4	ARRZ

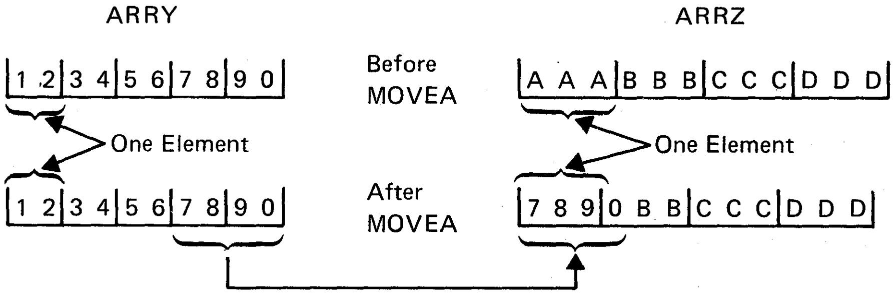
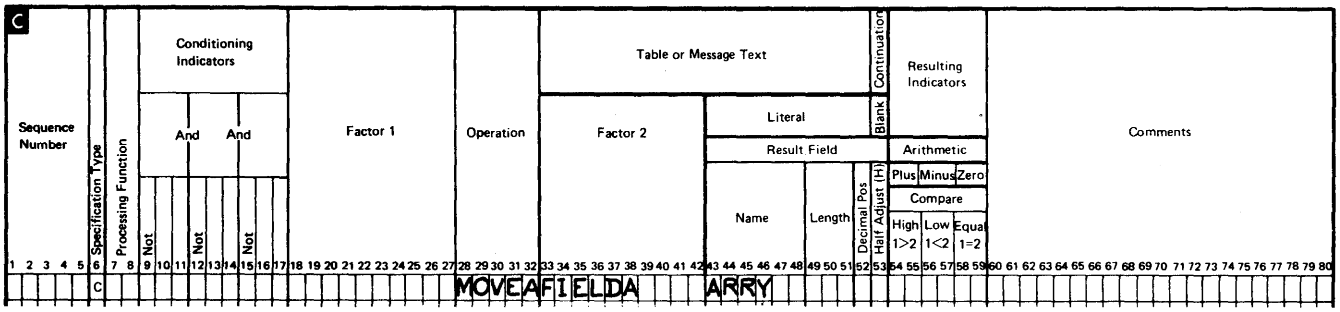


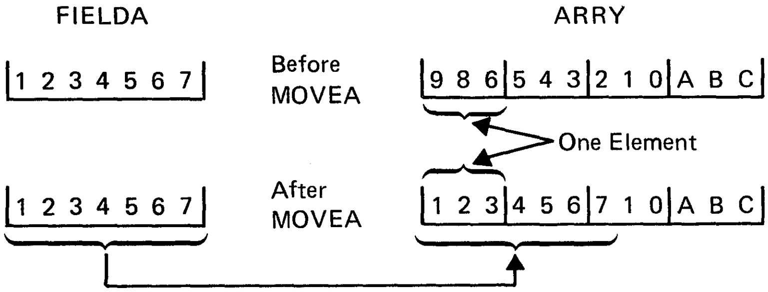
Figure 7-5 (Part 2 of 3). MOVEA Operation

Example: Field-to-array move.
No indexing on array.

S/36 WSU Processing Specifications

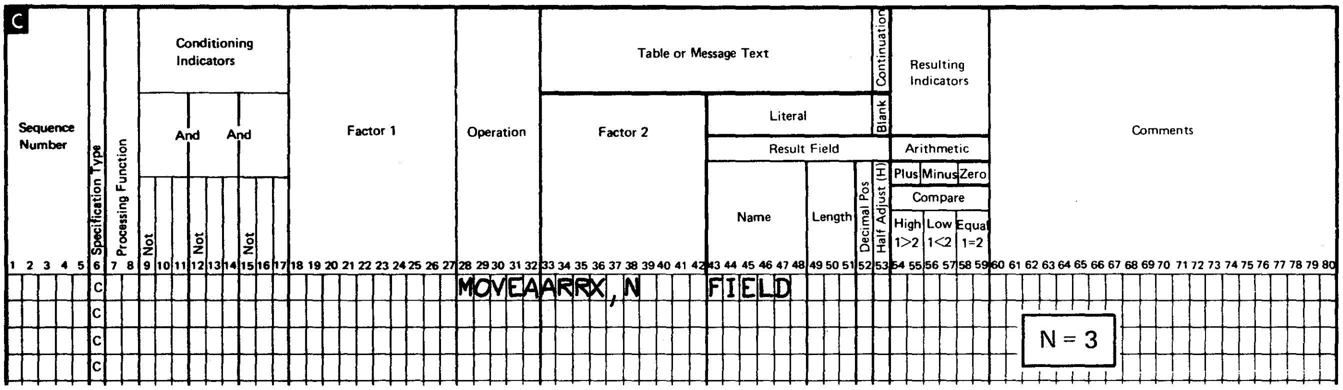


```
     C                     MOVEAFIELDA    ARRY
```



Example: Array-to-field move.
Variable indexing.

S/36 WSU Processing Specifications



```
     C                     MOVEAARRX,N    FIELD
     C
     C
     C
```

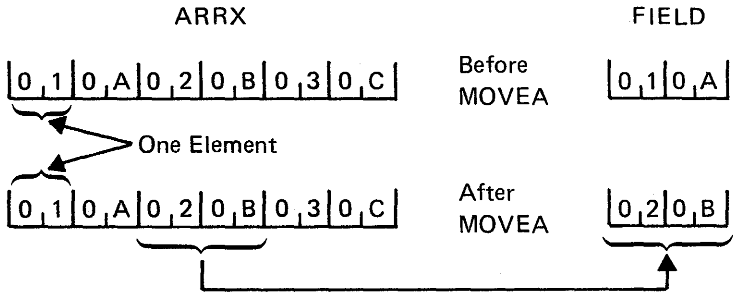
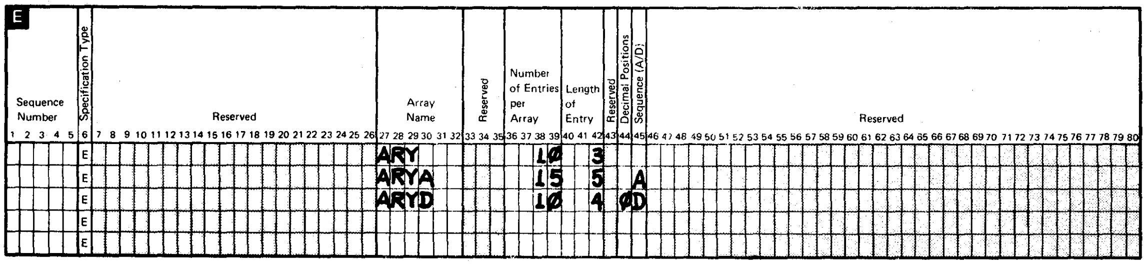
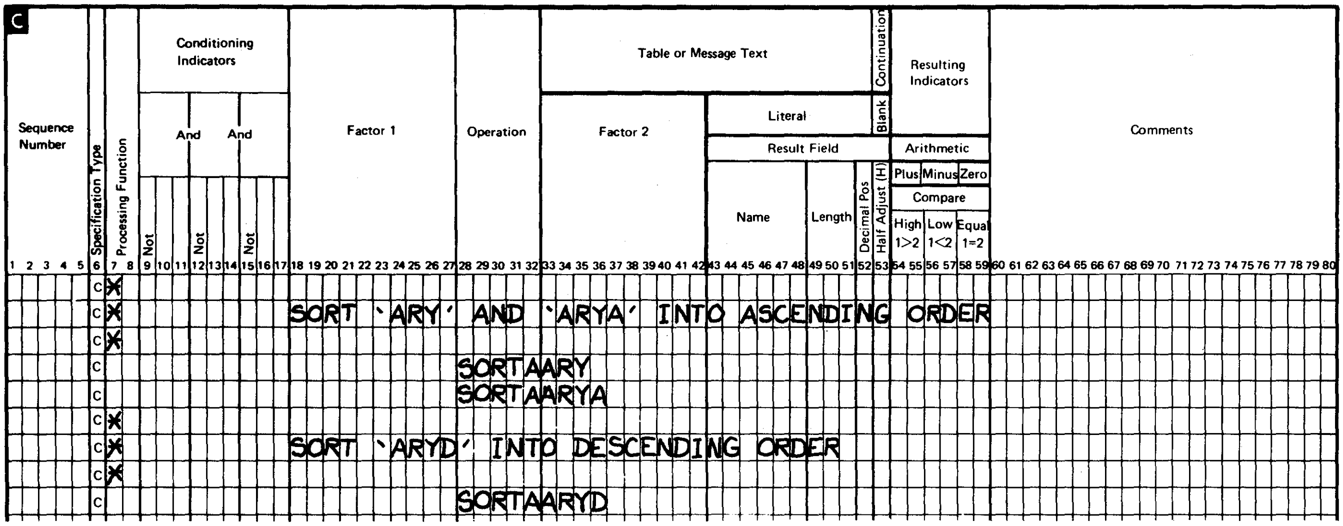


Figure 7-5 (Part 3 of 3). MOVEA Operation

Array Specifications



```
     E                    ARY         10  3
     E                    ARYA        15  5 A
     E                    ARYD        10  40D
     E
     E
```



```
     C*
     C*          SORT 'ARY' AND 'ARYA' INTO ASCENDING ORDER
     C*
     C                      SORTAARY
     C                      SORTAARYA
     C*
     C*          SORT 'ARYD' INTO DESCENDING ORDER
     C*
     C                      SORTAARYD
```

Figure 7-6. SORTA Operation

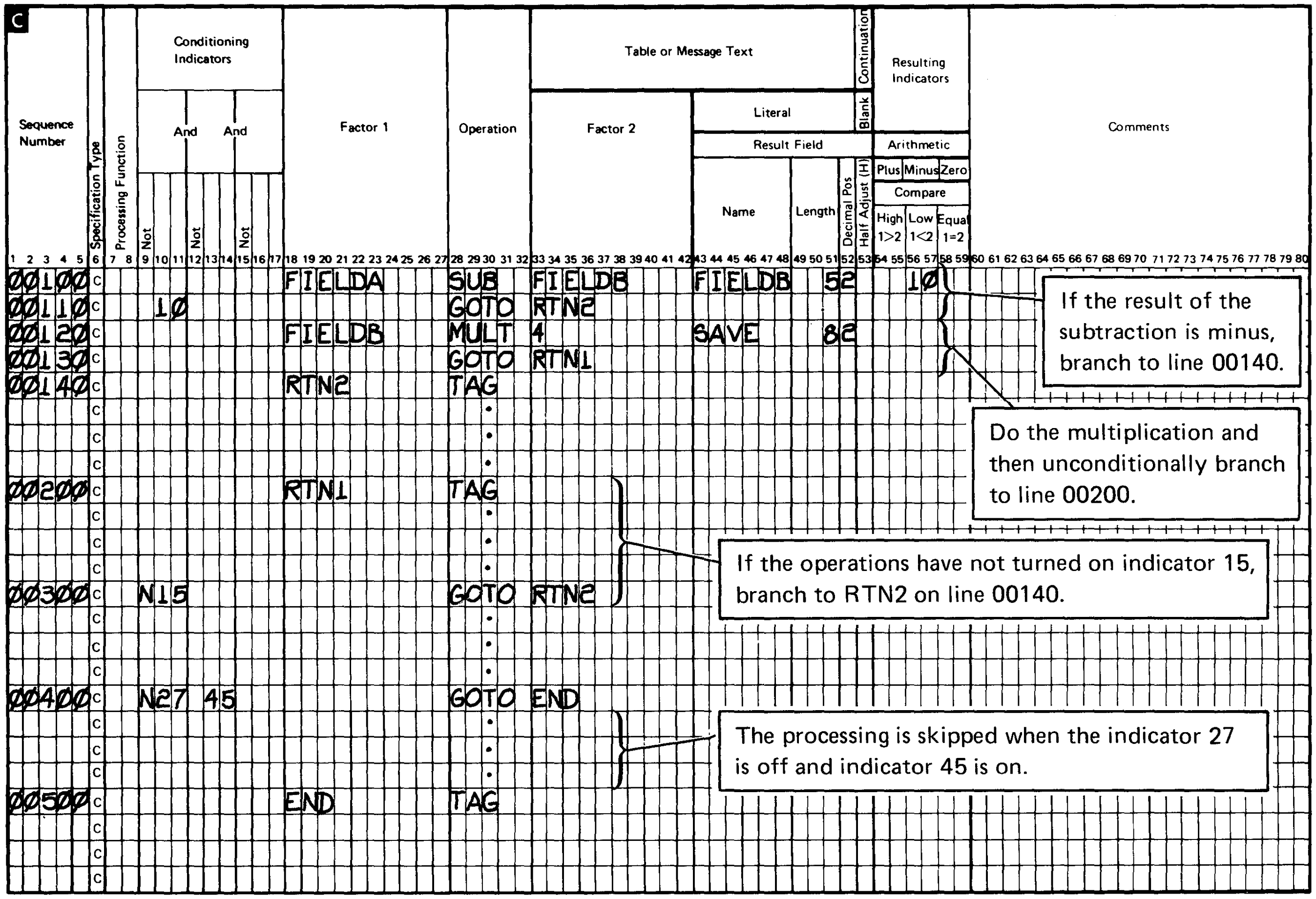
In this figure the array ARY is sorted into ascending order because no entry is specified for sequence (column 45) in the array specifications. ARYA is sorted into ascending order because column 45 of the array specifications contains A; ARYD is sorted into descending order because column 45 contains D.

Branching Operations

Operations run in the order in which you code them on the C-specifications. Sometimes, however, you might want to skip several operations when certain conditions occur or you might want to do several operations repeatedly. The GOTO and TAG operations allow branching within C-specifications for a display, for a subroutine, or for a processing level.

- GOTO (go to) allows you to skip instructions and go to a TAG or ENDSR line. You can use GOTO to branch to a previous operation or to an operation that follows the GOTO operation. You cannot use GOTO to branch outside of C-specifications for a display, subroutine, or processing level.
- TAG (tag) names the line that a GOTO operation can branch to.

Refer to Figure 7-7 for examples of coding GOTO and TAG.



Sequence Number	Specification Type	Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Pos	Resulting Indicators
00100	C		FIELDA	SUB	FIELDB	FIELDB	5	2	10
00110	C	10		GOTO	RTN2				
00120	C		FIELDB	MULT	4	SAVE	8	2	
00130	C			GOTO	RTN1				
00140	C		RTN2	TAG					
	C			.					
	C			.					
	C			.					
00200	C		RTN1	TAG					
	C			.					
	C			.					
	C			.					
00300	C	N15		GOTO	RTN2				
	C			.					
	C			.					
	C			.					
00400	C	N27 45		GOTO	END				
	C			.					
	C			.					
	C			.					
00500	C		END	TAG					
	C								
	C								
	C								

If the result of the subtraction is minus, branch to line 00140.

Do the multiplication and then unconditionally branch to line 00200.

If the operations have not turned on indicator 15, branch to RTN2 on line 00140.

The processing is skipped when the indicator 27 is off and indicator 45 is on.

Figure 7-7. GOTO and TAG Operations

Comparing and Testing Operations

The operations for comparing and testing are COMP and RANGE. Fields do not change because of these operations.

- COMP (compare) compares factor 1 with factor 2.
- RANGE (range check) compares factor 1 with both factor 2 and the result field.

Comparing Factor 1 to a Field or Literal

The COMP operation allows a field or literal to be compared with one or more fields. As a result of the compare, indicators turn on as follows:

High: Factor 1 is greater than factor 2

Low: Factor 1 is less than factor 2

Equal: Factor 1 equals factor 2

Factor 1 and factor 2 must be the same data type; both alphabetic, both numeric, or both ideographic. Coding a High or Low compare operation with ideographic characters will produce unpredictable results (refer to *Ideographic Sort Guide*, SC09-1056 for details).

The data is automatically aligned before the compare. If the data is alphameric or ideographic, it is aligned to the leftmost character. If one factor is shorter, the unused positions fill with blanks (Figure 7-8). The maximum length for compared alphameric data is 256 characters.

If the data is numeric, it aligns according to the decimal point. Any missing digits fill with zeros (Figure 7-9). The maximum length of compared numeric data is 15 digits.

Note: The actual alignment and filling with blanks or zeros is done in a separate work area. Factor 1 is not changed.

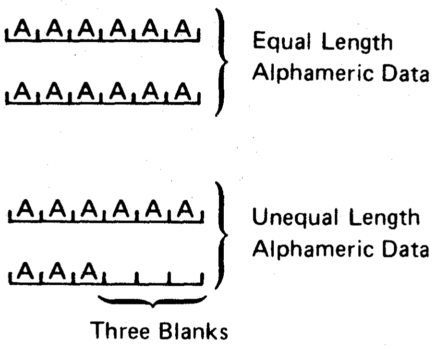


Figure 7-8. Comparing Alphameric Fields

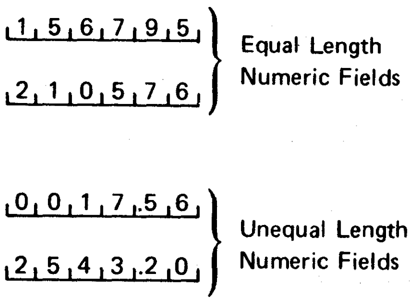


Figure 7-9. Comparing Numeric Fields

Figure 7-10 shows C-specifications for compare operations.

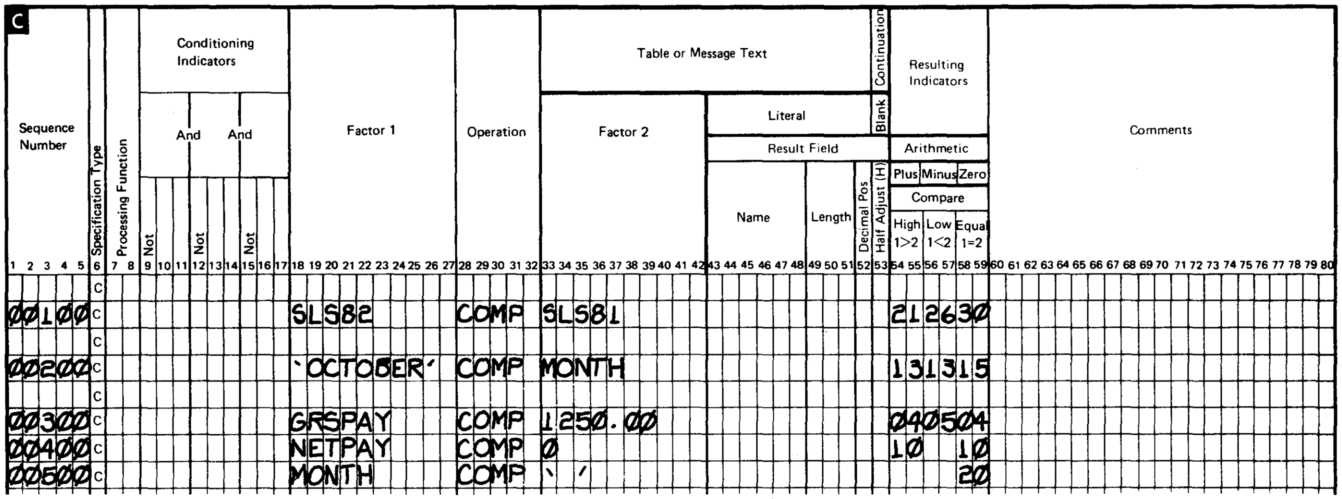
Line 00100: The contents of the field SLS82 (1982 sales) are compared with the contents of SLS81. Indicators 21, 26, and 30 are set off as the result of the COMP operation. If 1982 sales exceed 1981 sales, resulting indicator 21 turns on; if 1982 sales are less than 1981 sales, indicator 26 turns on; if the 2 years had equal sales, indicator 30 turns on.

Line 00200: The alphameric constant OCTOBER is compared with the contents of the field named MONTH (which must also be defined as alphameric). Indicators 13 and 15 are set off as the result of the COMP operation. If the MONTH field does not contain the word OCTOBER, indicator 13 turns on; if it does, indicator 15 turns on after the compare operation.

Line 00300: The contents of the field named GRSPAY (which must be numeric) decimal-aligns with numeric constant 1250.00. Indicators 04 and 05 are set off as the result of the COMP operation. If the value in field GRSPAY is greater than or equal to 1250.00, indicator 04 turns on; if the value in field GRSPAY is less than 1250.00, indicator 05 turns on.

Line 00400: The contents of the field NETPAY (which must be numeric) is decimal-aligned with numeric constant 0, and then compared to it. Indicator 10 is set off as the result of the COMP operation. If NETPAY is greater than 0 or equal to 0, indicator 10 turns on.

Line 00500: The contents of the field MONTH (which must be alphameric) is compared with a blank. Indicator 20 is set off as the result of the COMP operation. If MONTH is blank, indicator 20 turns on.



Sequence Number	Specification Type	Factor 1	Operation	Factor 2	Resulting Indicators High 1>2	Low 1<2	Equal 1=2
	C						
00100	C	SLS82	COMP	SLS81	21	26	30
	C						
00200	C	'OCTOBER'	COMP	MONTH	13	13	15
	C						
00300	C	GRSPAY	COMP	1250.00	04	05	04
00400	C	NETPAY	COMP	0	10		10
00500	C	MONTH	COMP	' '			20

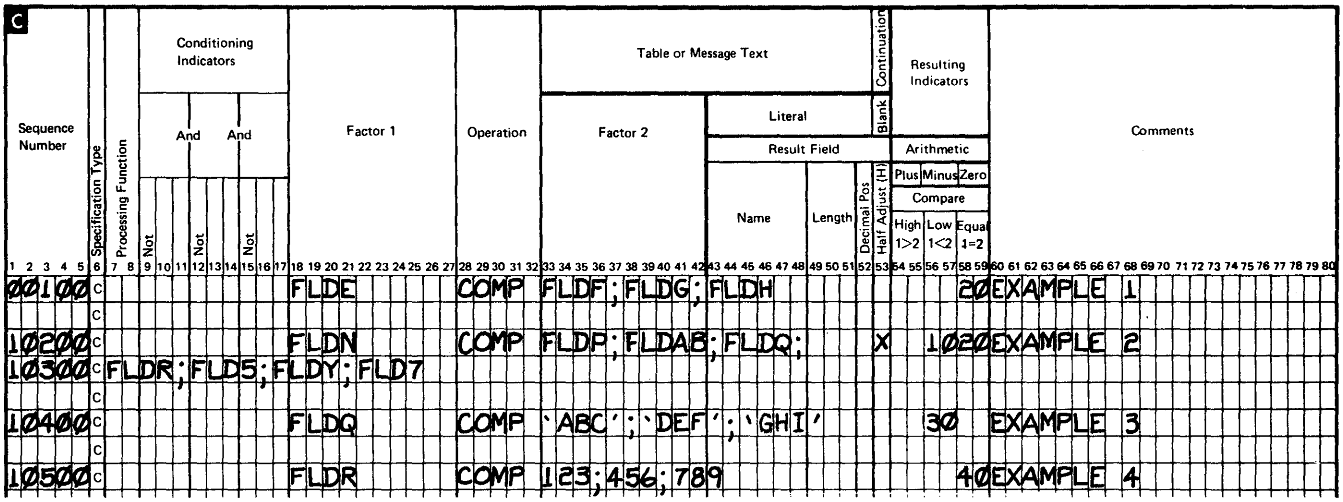
Figure 7-10. Compare (COMP) Operations

Comparing Factor 1 to a Table

The COMP operation can compare factor 1 to an array or a table of values (element by element) that is defined in columns 33 through 52. The following coding rules apply:

- Factor 1 must be the same data type (alphabetic, numeric, or ideographic) as the table entries. Coding a High or Low compare operation with ideographic characters will produce unpredictable results (refer to *Ideographic Sort Guide*, SC09-1056 for details).
- Table entries must either be field names or literals and cannot be a combination of the two. Table entries must be of one type; alphameric and numeric entries cannot be in the same table. Ideographic data is treated as alphameric data.
- If factor 1 and a table entry are different lengths, WSU uses the length of the longer value. Alphameric and ideographic data is aligned on the leftmost character, and unused positions in the shorter value are filled with blanks. Numeric data is aligned on the decimal point. Missing digits in the shorter value are filled with zeros.
- A semicolon (;) separates table elements.
- Continue a table to columns 7 through 52 of a second specification line by coding any character except a blank in column 53 of the first specification line. For a continued table, the first line must end with a semicolon. The second line of the table must begin with a left-adjusted table entry. Refer to the second COMP example in Figure 7-11 for an example of continuing a table.

  A table of values cannot be continued to a third line. Do not begin the second line of the table with an asterisk (*) in column 7 because WSU treats such lines as comments.
- A table of literals cannot have more than 140 positions. To calculate the number of positions in a table, multiply the number of elements by the following:
  - For alphameric literals, the number of characters (excluding apostrophes) in the longest one.
  - For numeric literals, the maximum number of digits to the left of the decimal point plus the maximum number of digits to the right of the decimal point. This sum must not exceed 15.
  - For ideographic literals, two plus twice the number of characters in the longest one.
- A table of field names can have as many as 10 names.



```
00100C           FLDE      COMP FLDF;FLDG;FLDH           20EXAMPLE 1
     C
10200C           FLDN      COMP FLDP;FLDAB;FLDQ;    X  1020EXAMPLE 2
10300CFLDR;FLD5;FLDY;FLD7
     C
10400C           FLDQ      COMP 'ABC';'DEF';'GHI'      30  EXAMPLE 3
     C
10500C           FLDR      COMP 123;456;789              40EXAMPLE 4
```

Figure 7-11. Examples of COMP Operations Using Tables

As a result of the comparison, the low and equal indicators are set off; then indicators turn on as follows:

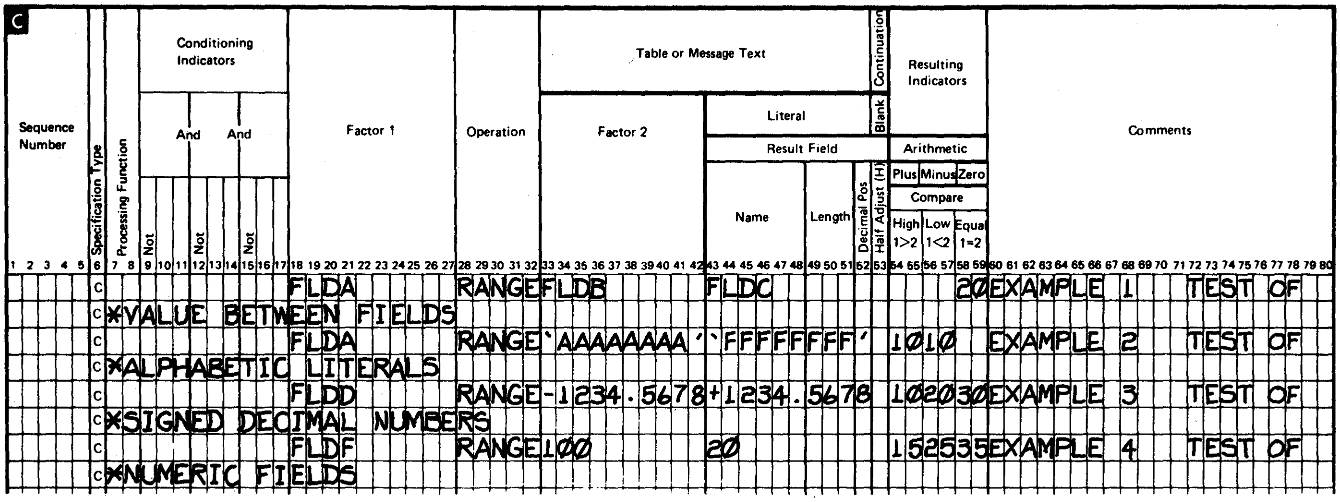
High: Not used.

Low: The value of the field named by factor 1 is not an element in the table.

Equal: The value of the field named by factor 1 is an element in the table.

Range Operation

The RANGE operation can be used to test if a field or literal has a value that is between two other values (compares factor 1 with both factor 2 and the result field). Refer to Chapter 18, *Operation Codes*, for more about the use of indicators in the RANGE operation. Figure 7-12 shows examples of the RANGE operation.



```
     C           FLDA      RANGEFLDB      FLDC           20EXAMPLE 1   TEST OF
     C*VALUE BETWEEN FIELDS
     C           FLDA      RANGE'AAAAAAAA''FFFFFFF'  1010  EXAMPLE 2   TEST OF
     C*ALPHABETIC LITERALS
     C           FLDD      RANGE-1234.5678+1234.5678 102030EXAMPLE 3   TEST OF
     C*SIGNED DECIMAL NUMBERS
     C           FLDF      RANGE100       20         152535EXAMPLE 4   TEST OF
     C*NUMERIC FIELDS
```

Figure 7-12. Examples of the RANGE Operations

Debugging Operation

The operation for debugging (a means of detecting errors) a WSU program while it runs is DEBUG. If DEBUG-YES is specified in the procedure that calls the program, WSU shows a display of debugging information when the DEBUG operation is encountered.

Chapters 9, 17, and 18 contain more information on debugging.

Indicator Operations (Setting On and Off)

The SETON and SETOF operations set the following indicators on or off:

AC

AE

EJ

ES

EW

IP

IS

JA-JN, JP-JY

KG-KL, KQ-KY

PG

RC

RP

RS

SA-SN, SP-SY

U1-U8

01-89

Refer to Chapter 3, *Reserved Fields and Indicators*, for a review of indicator usage.

Input and Output Operations

WSU provides the following operations to control input and output for files, to show messages on a display, to show a display, and to get the system time of day and system date:

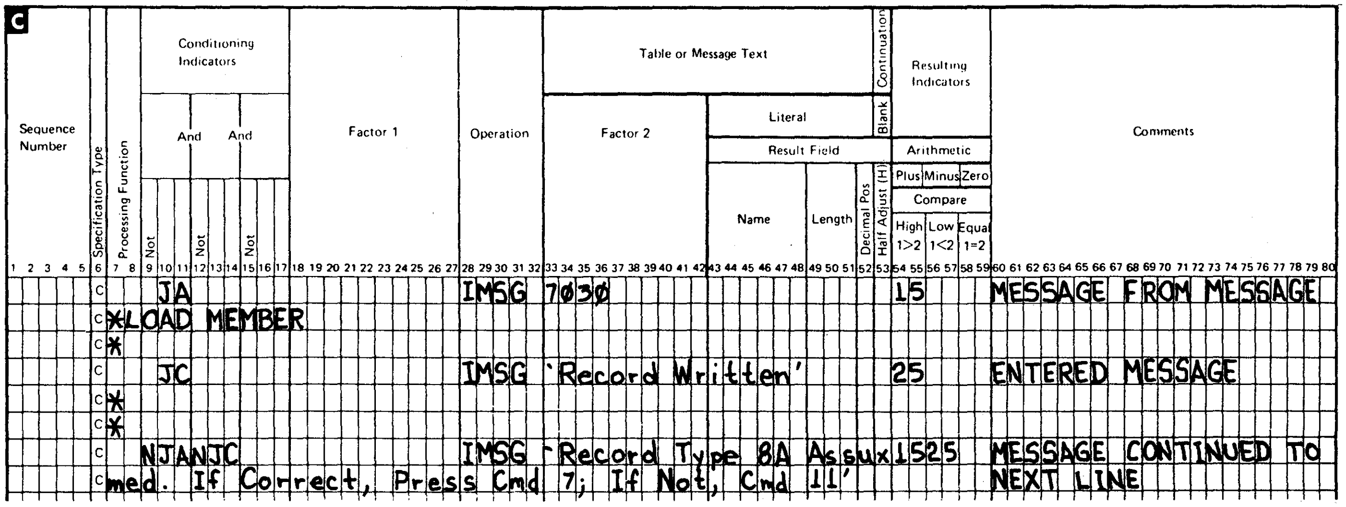
- GET (get a master file record) reads a record from the master file. The master file data can also be updated by the program.
- GETNH (get the next header record from the transaction file) reads the next logical header record from the transaction file up to the end of a chain of records.
- GETNR (get the next record from the transaction file) reads the next logical record from the transaction file up to the end of a chain of records.
- GETPH (get the previous header record from the transaction file) reads the previous logical header record from the transaction file up to the beginning of a chain of records. The record is read only; it is not displayed.

- GETPR (get the previous record from the transaction file) reads the previous logical record from the transaction file up to the beginning of a chain of records. The record is read only; it is not displayed.

For a GETNH, GETNR, GETPH, or GETPR operation, if a transaction file is not specified in a WSU program, the operations cannot be coded. These operations cause terminal errors during program compiling.

- IMSG (display an information message) causes a user-defined message to be shown on the bottom line of the current display.

Figure 7-13 shows examples of the IMSG operation.

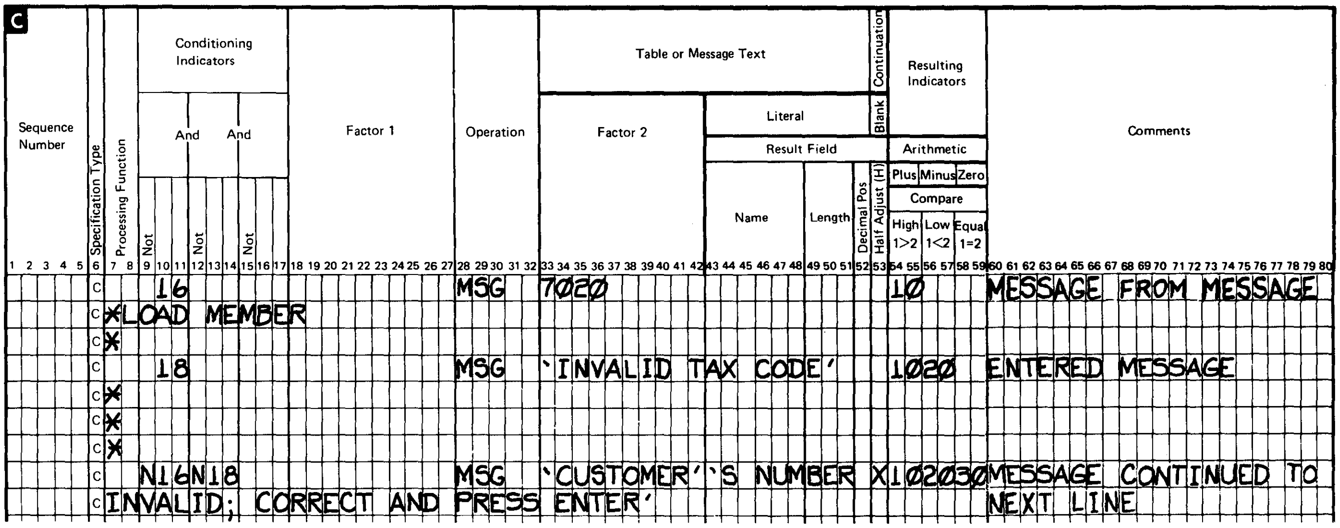


```
     C   JA                IMSG 7030                 15    MESSAGE FROM MESSAGE
     C*LOAD MEMBER
     C*
     C   JC                IMSG 'Record Written'     25    ENTERED MESSAGE
     C*
     C*
     C  NJANJC             IMSG 'Record Type 8A Assux1525  MESSAGE CONTINUED TO
     Cmed. If Correct, Press Cmd 7; If Not, Cmd 11'        NEXT LINE
```

Figure 7-13. Examples of the IMSG Operation

- MSG (display a diagnostic message) interrupts the processing of a display and causes the current display to be immediately shown with a user message on the bottom line of the display. The message overlays any data that is on the bottom line of the display.

Figure 7-14 shows examples of the MSG operation.



```
     C   16                MSG  7020                 10    MESSAGE FROM MESSAGE
     C*LOAD MEMBER
     C*
     C   18                MSG  'INVALID TAX CODE'   1020  ENTERED MESSAGE
     C*
     C*
     C*
     C  N16N18             MSG  'CUSTOMER''S NUMBER X102030MESSAGE CONTINUED TO
     CINVALID; CORRECT AND PRESS ENTER'                    NEXT LINE
```

Figure 7-14. Examples of the MSG Operation

- PUT (write a transaction file or master file record) writes one record to the transaction file or a master file.

  If the PUT operation does not specify a record type, WSU refers to the most recent get operation that occurred for the file since the most recent mode change and assumes that the record type recognized and saved from this operation is the record type for the PUT operation.

  The following summarizes the PUT operation:

Transaction File Record PUT

Mode	Record Type in Columns 54 and 55	Blanks in Columns 54 and 55 (see notes)
Enter	Adds a record to the file	Updates the last record read using GETNR, GETNH, GETPR, GETPH
Review	Updates the displayed record	Updates the last record read using GETNR, GETNH, GETPR, GETPH
Insert	Inserts a record after the displayed record	Updates the last record read using GETNR, GETNH, GETPR, GETPH

Notes:

1. If a PUT operation without a record type runs and one of the GET operations has not occurred for a record since the most recent mode change, WSU does *one* of the following:
   - Sets on the error indicator coded on the T-specification and places an error code in *ERROR.
   - Displays WSU message 0803 and an error code at the system console if an error indicator has not been coded on the T-specification.
2. If the PUT operation occurs after a Roll or Page Record Group command key is used to start a review function and a record type is satisfied, the transaction file record that was first found to satisfy the request is the record that is updated by the PUT operation.

Master File Record PUT

Mode	Blanks in Columns 54 and 55 (see note)
Enter	Updates the last record read using GET
Review	Updates the last record read using GET
Insert	Updates the last record read using GET

Note: If a PUT operation without an ID runs and a get operation has not occurred for a record since the most recent mode change, WSU does *one* of the following:

- Sets on the error indicator coded on the M-specification and places an error code in *ERROR.
- Displays WSU message 0803 and an error code at the system console if an error indicator has not been coded on the M-specification.
- PUTN (add a record to a master file) adds one record to a master file.
- PUTS (show a display) causes the display named in factor 2 to be selected for processing. During the processing of a display, the programmed processing can be modified by means of the PUTS operation. See *Using PUTS in Special Processing Level Calculations* later in this chapter.
- TIME (get the time of day and date) moves the system time of day or the system time of day and the system date into the result field.

  The format of the time of day is hhmmss, where hh specifies hours of the 24-hour clock, mm specifies minutes, and ss specifies seconds.

  The format of the system date is determined by the system configuration. This format can be mmddyy, ddmmyy, or yymmdd, where mm specifies the month, dd specifies the day, and yy specifies the last two digits of the year.

  The system date placed in the result field might not match the program date that the UDATE field contains.

  Refer to the *System Reference* manual for a description of the system date, program date, and system time of day.

Move Operations

The move operations are MOVE and MOVEL (move left). These operations move (copy) part or all of factor 2 to the result field. Factor 2 does not change, and factor 1 and columns 54 through 59 must be blank.

- MOVE (move) moves all or part of factor 2 into the result field; moving begins with the rightmost character.

  The MOVE operation can be used to change numeric data into alphameric data and alphameric data into numeric data.

  Figure 7-15 shows examples of the MOVE operation.

- MOVEL (move left) moves all or part of factor 2 into the result field; moving begins with the leftmost character.

  Figure 7-16 summarizes the MOVEL operation.

Factor 2 and Result Field Same Length

	Factor 2		Result Field	
Alphameric	P H 4 S N	Before MOVE	5 6 7 8 4	Alphameric
	P H 4 S N	After MOVE	P H 4 S N	
Alphameric	P H 4 S N	Before MOVE	5 6 7 8 4	Numeric
	P H 4 S N	After MOVE	7 8 4 2 $\overset{-}{5}$	
Numeric	7 8 4 2 5	Before MOVE	1 3 3 5 6	Numeric
	7 8 4 2 5	After MOVE	7 8 4 2 5	
Numeric	7 8 4 2 5	Before MOVE	A L T 5 F	Alphameric
	7 8 4 2 5	After MOVE	7 8 4 2 5	

Factor 2 Shorter Than Result Field

	Factor 2		Result Field	
Alphameric	P H 4 S N	Before MOVE	1 2 3 4 5 6 7 8 $\overset{+}{4}$	Alphameric
	P H 4 S N	After MOVE	1 2 3 4 P H 4 S N	
Alphameric	P H 4 S N	Before MOVE	1 2 3 4 5 6 7 8 $\overset{+}{4}$	Numeric
	P H 4 S N	After MOVE	1 2 3 4 7 8 4 2 $\overset{-}{5}$	($\overset{-}{5}$ = letter N)
Numeric	1 2 7 8 4 2 5	Before MOVE	1 2 3 4 5 6 7 8 $\overset{+}{9}$	Numeric
	1 2 7 8 4 2 5	After MOVE	1 2 1 2 7 8 4 2 $\overset{+}{5}$	
Numeric	1 2 7 8 4 2 5	Before MOVE	A C F G P H 4 S N	Alphameric
	1 2 7 8 4 2 5	After MOVE	A C 1 2 7 8 4 2 5	

Factor 2 Longer Than Result Field

	Factor 2		Result Field	
Alphameric	A C E G P H 4 S N	Before MOVE	5 6 7 8 4	Alphameric
	A C E G P H 4 S N	After MOVE	P H 4 S N	
Alphameric	A C E G P H 4 S N	Before MOVE	5 6 7 8 $\overset{+}{4}$	Numeric
	A C E G P H 4 S N	After MOVE	7 8 4 2 $\overset{-}{5}$	
Numeric	1 2 7 8 4 2 5	Before MOVE	5 6 7 4 8	Numeric
	1 2 7 8 4 2 5	After MOVE	7 8 4 2 5	
Numeric	1 2 7 8 4 2 5	Before MOVE	P H 4 S N	Alphameric
	1 2 7 8 4 2 5	After MOVE	7 8 4 2 5	

$\overset{+}{4}$ = letter D

$\overset{-}{5}$ = letter N

Figure 7-15. MOVE Operation

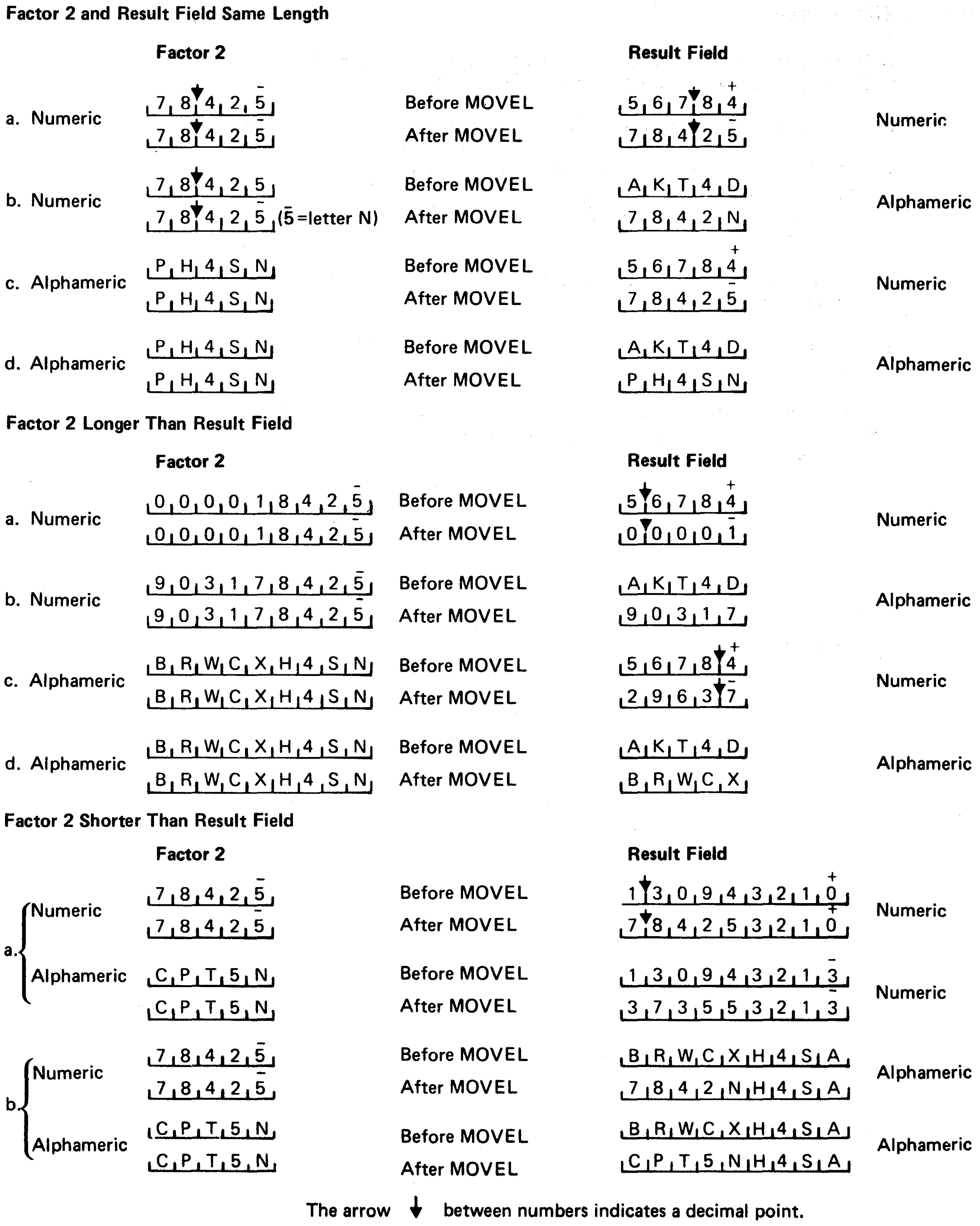


Figure 7-16. MOVEL Operation

Subroutine Operations

WSU provides the following operations for defining the beginning, end, and priority of subroutines in a WSU program:

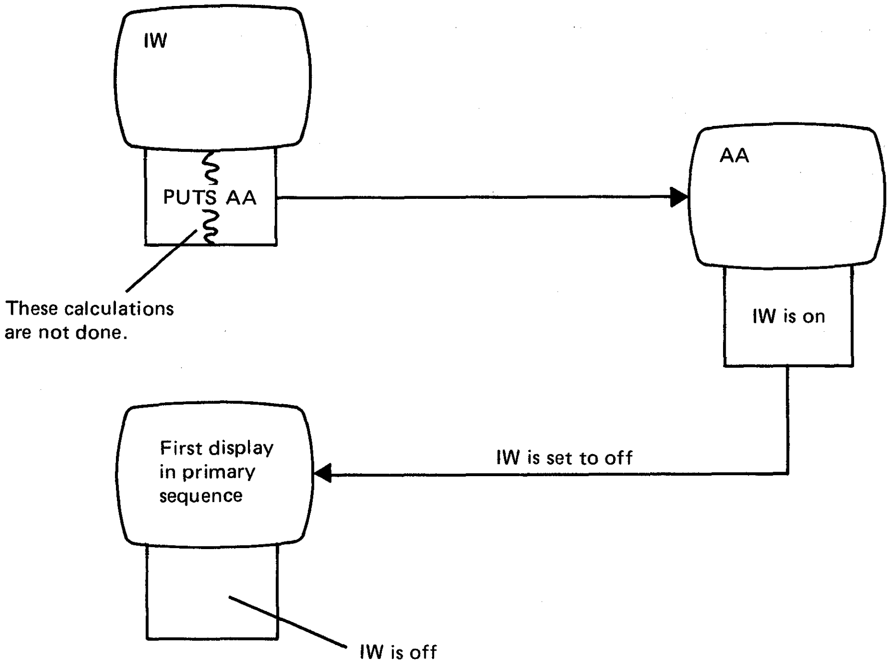
- BEGSR (begin subroutine) labels the beginning of the subroutine.
- ENDSR (end subroutine) defines the end of a subroutine and causes control to return to the instruction after the EXSR operation that called the subroutine. BEGSR and ENDSR operation must be coded in pairs.
- EXSR (execute subroutine) branches to and runs the subroutine named in factor 2.
- PRTY (priority) assigns a run priority (expected frequency of use) to the subroutine.

Subroutines must be coded at the end of a WSU program. Any WSU operation except EXSR can be coded in a subroutine. See *Subroutines* in this chapter.

Using PUTS in Special Processing Level Calculations

If you issue a PUTS operation from a special processing level (IJ, IW, ES, EW, or EJ), the processing level indicator remains on during the processing of the PUTS display. When the calculations for the PUTS display have all been done, WSU sets the indicator off and selects the next processing level display or the first display in the primary sequence. All calculations that followed the PUTS operation in the processing level are not done. If the display shown by the PUTS operation is normally part of a sequence of displays, the other displays in the sequence are ignored. Only the processing for the display shown by the PUTS operation is performed before the next processing level display or the first display of the primary sequence is selected. If the Bypass Display command key (Cmd 2) is pressed while the display shown by the PUTS operation is displayed, any processing for the display is ignored, and the next processing level display or the first display of the primary sequence is selected.

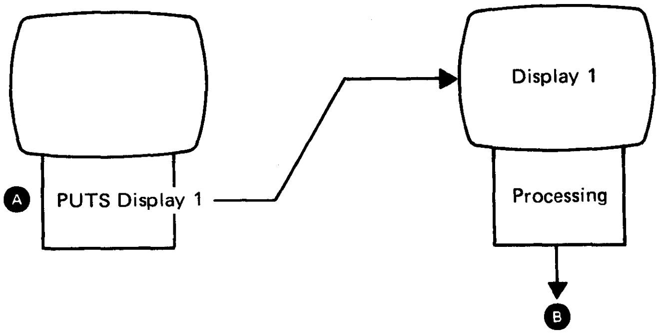
The following chart shows what happens when a PUTS operation is used in IW processing.



Examples Using PUTS

The following examples, 1 through 8, illustrate various uses of the PUTS operation during an enter mode sequence.

Example 1: Use PUTS to show a display for which processing has been specified.



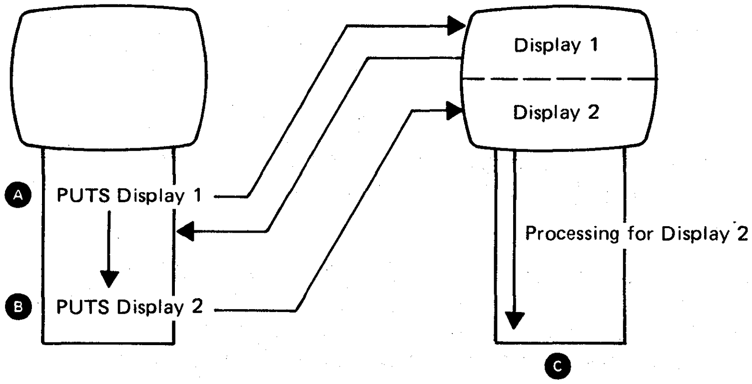
Ⓐ The PUTS operation causes Display 1 to appear.

When the operator enters Display 1, processing begins with its first C-specification.

Ⓑ Assume that Display 1 is in a sequence of displays. When the processing for Display 1 ends, WSU selects the next display in the sequence.

Example 2: Use PUTS to show a display (Display 1), for which processing has not been specified. Display 1 will be shown on the top half of the display screen. Assume that Reset Keyboard (columns 23 and 24) on the S-specification is N for this display. (Because the Reset Keyboard entry is N, Display 1 must be a nonsequenced display, which means that its S- and D-specifications must follow those S- and D-specifications that describe the sequenced displays in the WSU program.)

Use a second PUTS to show a display (Display 2) on the bottom half of the display screen. Assume that Display 2 has processing, does not have preprocessing, does not clear the display, and is in an enter mode sequence of displays.

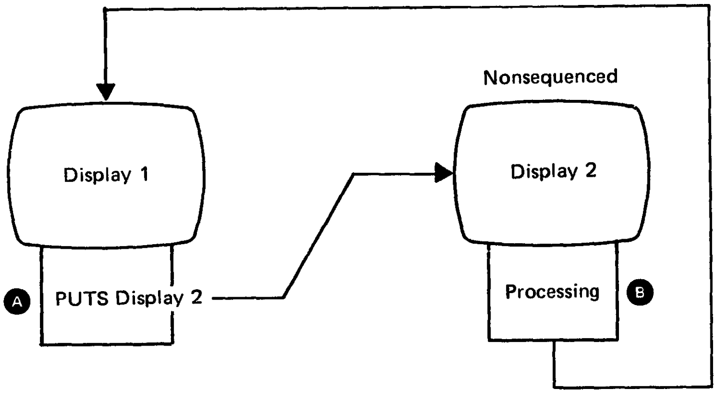


Ⓐ The first PUTS operation causes Display 1 to appear. Because Display 1 has neither processing specifications nor Reset Keyboard specified, control returns immediately to the operation that follows the PUTS for Display 1.

Ⓑ The second PUTS operation causes Display 2 to appear. Display 1 remains on the display screen. When the operator enters Display 2, processing begins with its first C-specification.

Ⓒ When the processing for Display 2 ends, WSU selects the next display in the sequence.

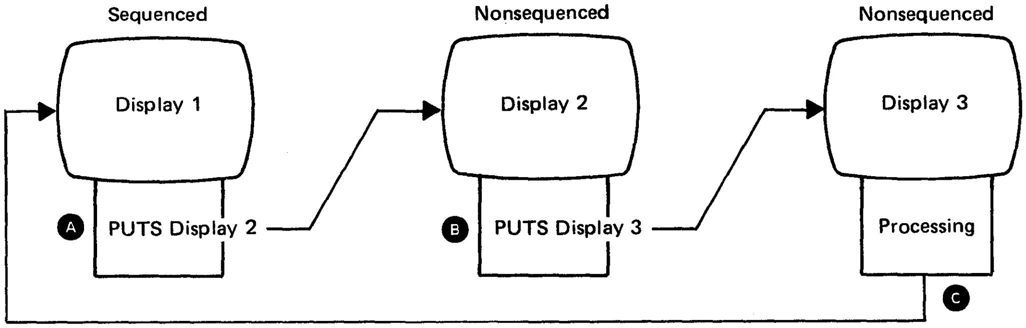
Example 3: Use PUTS to show a display that is not sequenced (Display 2) from a sequenced display (Display 1). Assume that neither Display 1 nor Display 2 has preprocessing and that both displays occur during an enter mode sequence.



Ⓐ The PUTS operation causes Display 2 to appear. When the operator enters Display 2, processing begins with its first C-specification.

Ⓑ When the processing for Display 2 ends, WSU shows Display 1.

Example 4: Use PUTS to show a display that is not sequenced (Display 3) from a display that is not sequenced (Display 2). Assume that Display 1 is in an enter mode sequence of displays and that these three displays do not have preprocessing.



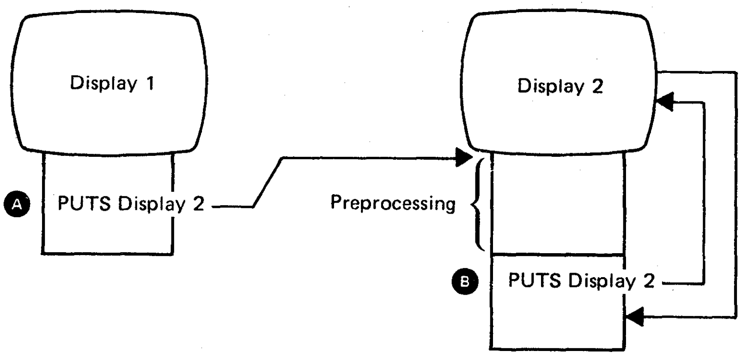
Ⓐ The PUTS operation from Display 1 causes Display 2 to appear. When the operator enters this display, processing begins with its first C-specification.

Ⓑ This PUTS operation from Display 2 causes Display 3 to appear. When the operator enters this display, processing begins with its first C-specification.

Ⓒ When the processing for Display 3 ends, WSU shows Display 1 (the most recent sequenced display).

Note: If a level indicator (such as ES, EW, or EJ) is on when processing for Display 3 ends, the display for that processing level appears next.

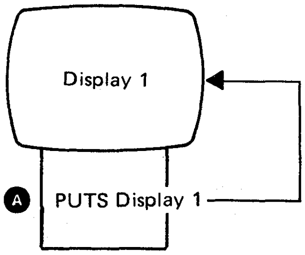
Example 5: Use PUTS to show a display that has preprocessing.



Ⓐ Processing begins immediately for Display 2 when the PUTS operation for Display 2 occurs. Display 1 can be a processing level display.

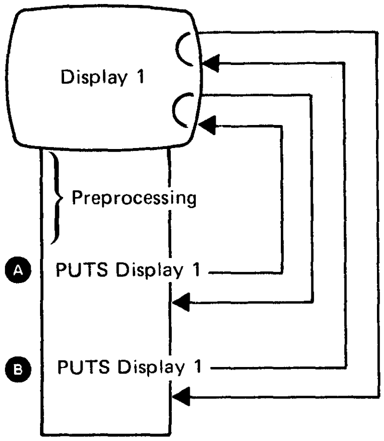
Ⓑ Display 2 appears when this PUTS occurs. After Display 2 appears, the operation that follows the PUTS operation runs. During review, insert, or delete modes, Display 2 does not have to support review, insert, or delete mode for the current record type. Also, during one of these modes, if the operator presses the Bypass Display command key (Cmd 2) in response to Display 2, the search for a display that supports review, insert, or delete mode for the current record type begins at the display following Display 2.

Example 6: Use PUTS to show again a display that does not have preprocessing.



Ⓐ The PUTS operation causes Display 1 to reappear. When the operator enters Display 1, processing begins with its first C-specification.

Example 7: Use PUTS to show a display that has preprocessing.

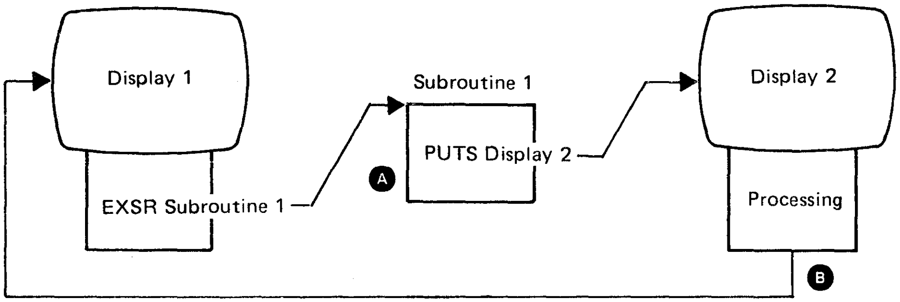


Ⓐ The first PUTS operation marks the end of the preprocessing and causes Display 1 to appear. When the operator enters Display 1, processing begins with the operation that follows the PUTS.

Ⓑ The second PUTS operation causes Display 1 to appear. When the operator enters Display 1, processing begins with the operation that follows this PUTS.

Note: If an MSG operation occurs between Ⓐ and Ⓑ, the preprocessing starts over. Consequently, the operator might have to respond to the display twice.

Example 8: Use PUTS in a subroutine to show a display that does not have preprocessing and does not repeat. Assume that Display 1 is in an enter mode sequence of displays and that Display 2 is not in a sequence of displays.



Ⓐ The PUTS operation causes Display 2 to appear. When the operator enters Display 2, processing begins with its first C-specification.

Ⓑ When the processing for Display 2 ends, WSU selects Display 1, the most recently shown sequenced display.

Chapter 8. Entering and Generating a WSU Program

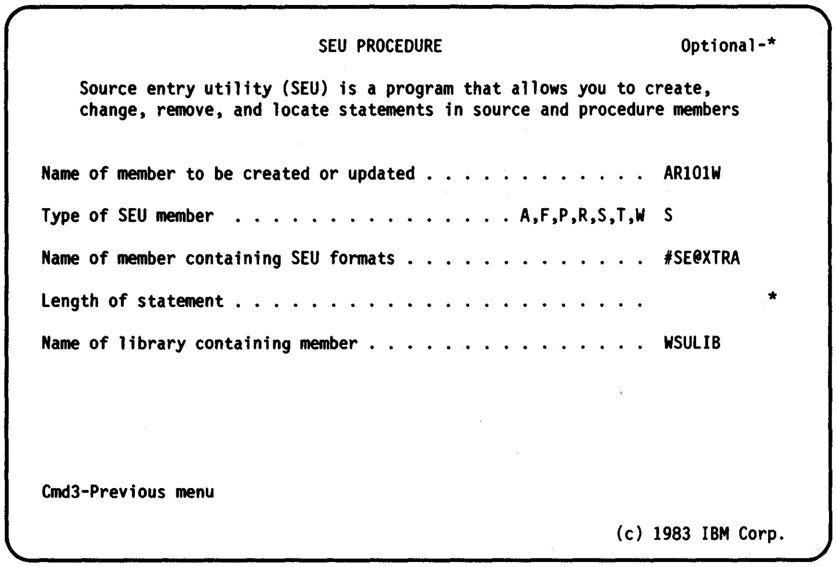
Once you have designed your program and coded it on the specification sheets, you are ready to enter the code into the system and generate (or compile) the program.

Entering The Code

You can enter (or change) the file definition and the source program specifications for your WSU program using SEU, just as you would for any other language/utility. The SEU method is described here briefly. You will, of course, need to be familiar with SEU or, at the very least, have a copy of the *SEU Guide* at hand when you are entering or changing your code.

Entering the File Definition

In response to the SEU procedure prompt:



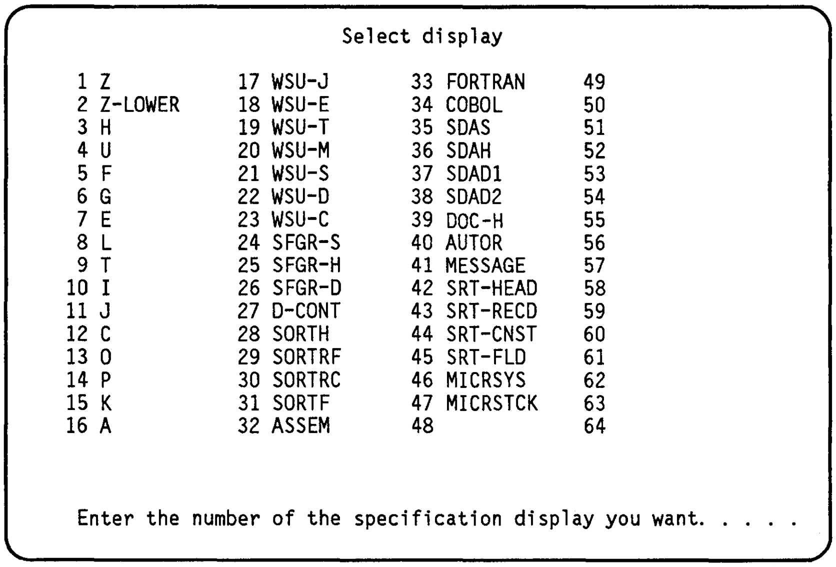
```
                         SEU PROCEDURE                       Optional-*

   Source entry utility (SEU) is a program that allows you to create,
   change, remove, and locate statements in source and procedure members


Name of member to be created or updated . . . . . . . . . . . .  AR101W

Type of SEU member  . . . . . . . . . . . . . . . A,F,P,R,S,T,W  S

Name of member containing SEU formats . . . . . . . . . . . . .  #SE@XTRA

Length of statement . . . . . . . . . . . . . . . . . . . . . .            *

Name of library containing member . . . . . . . . . . . . . . .  WSULIB




Cmd3-Previous menu

                                                              (c) 1983 IBM Corp.
```

you must enter:

- The name of the member to be created/changed. Remember that the file definition must be a separate member from the source program; the name must be the one you specify on the T- or M-specification.
- The type of member it is to be. You can use member type S (for source) for your source program if you don't want syntax verification. If you choose member type R (for RPG), SEU checks syntax as you enter the specifications; however, any special considerations for WSU will not be checked.
- The name of the library to contain the member.

The other options have defaults or are not required.

SEU offers these specification display types to select from:



```
                              Select display

   1 Z            17 WSU-J       33 FORTRAN     49
   2 Z-LOWER      18 WSU-E       34 COBOL       50
   3 H            19 WSU-T       35 SDAS        51
   4 U            20 WSU-M       36 SDAH        52
   5 F            21 WSU-S       37 SDAD1       53
   6 G            22 WSU-D       38 SDAD2       54
   7 E            23 WSU-C       39 DOC-H       55
   8 L            24 SFGR-S      40 AUTOR       56
   9 T            25 SFGR-H      41 MESSAGE     57
  10 I            26 SFGR-D      42 SRT-HEAD    58
  11 J            27 D-CONT      43 SRT-RECD    59
  12 C            28 SORTH       44 SRT-CNST    60
  13 O            29 SORTRF      45 SRT-FLD     61
  14 P            30 SORTRC      46 MICRSYS     62
  15 K            31 SORTF       47 MICRSTCK    63
  16 A            32 ASSEM       48             64


    Enter the number of the specification display you want. . . . .
```

You use display type F (number 5) for the F-specification(s) and display type I (number 10) and J (number 11) for the I-specifications.

Storing the File Definition

The F-specifications and I-specifications for a given *file* must be together in the same source member. However, a file definition for a *program* can be constructed in a number of ways:

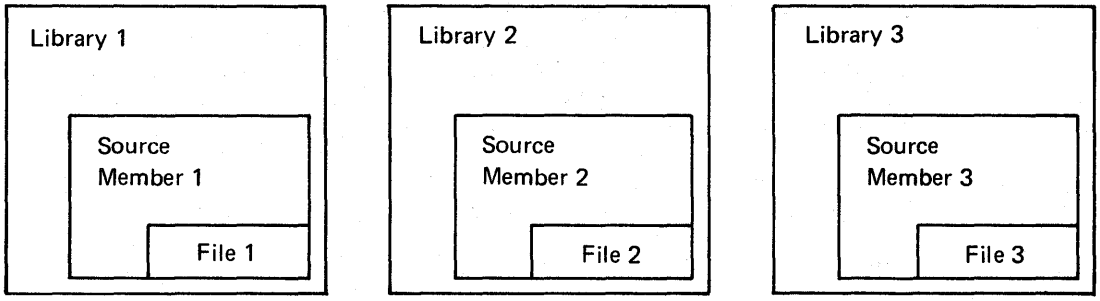
- One source member can contain the F- and I-specifications for *all* of the files that a program uses.

  If specifications for more than one file are included they can be arranged:

```
F FILE1                     F FILE1
F FILE2                     I FILE1
F FILE3                     I
I FILE1                     I
I                           I
I                           F FILE2
I                           I FILE2
I FILE2          or         I
I                           I
I                           I
I                           F FILE3
I FILE3                     I FILE3
I                           I
I                           I
I                           I
```

- Multiple source members in multiple libraries can contain F- and I-specifications for *different* files used by a program.

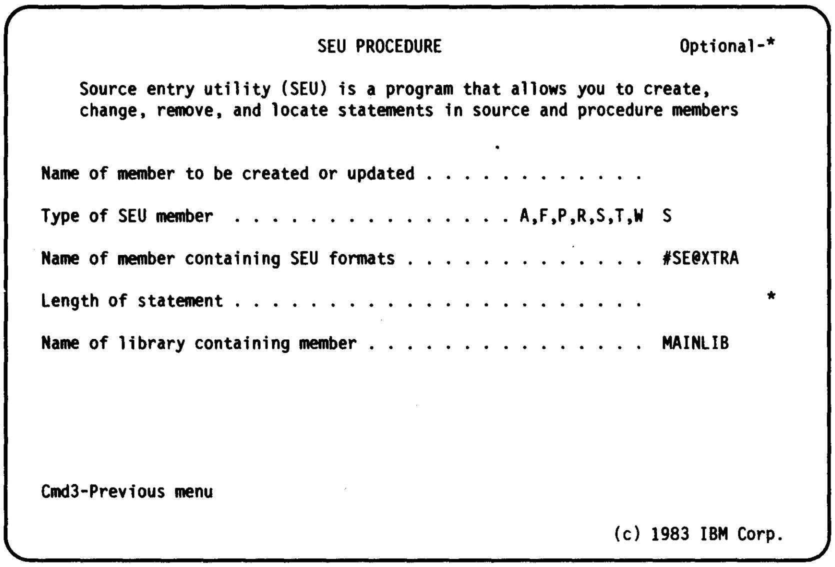
  For example:



- The F- and I-specifications for a WSU application program can be part of an RPG II program source member.

Entering the Source Program

In response to the SEU procedure prompt:



```
                         SEU PROCEDURE                      Optional-*

    Source entry utility (SEU) is a program that allows you to create,
    change, remove, and locate statements in source and procedure members

Name of member to be created or updated . . . . . . . . . . . .

Type of SEU member . . . . . . . . . . . . . . . A,F,P,R,S,T,W  S

Name of member containing SEU formats . . . . . . . . . . . . . #SE@XTRA

Length of statement . . . . . . . . . . . . . . . . . . . . . .            *

Name of library containing member . . . . . . . . . . . . . . . MAINLIB




Cmd3-Previous menu
                                                       (c) 1983 IBM Corp.
```

you must enter:

- The name of the member to be created/changed. Remember that the source program must be a separate member from the file definition; the name must be the one you specify on the WSU procedure command.
- The type of member it is to be. If you choose member type W (for WSU), SEU selects some of the display specifications you need automatically; however, it does not verify WSU syntax. You can also choose member type S (for source) if you don't want SEU to select displays.
- The name of the library to contain the member. The name must be the one you specify on the WSU procedure command.

The other options have defaults or are not required.

The specifications in a WSU source program must be in the following order:

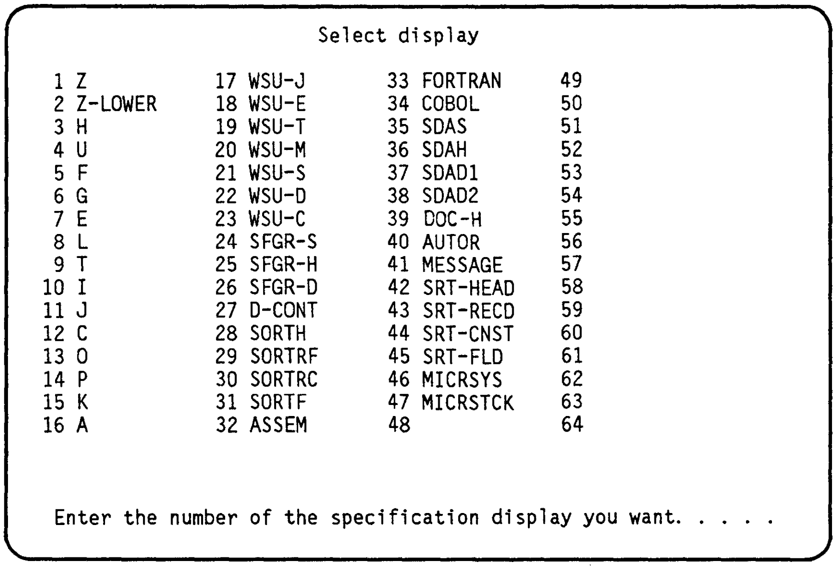
J (required)
E (if arrays are used)
.
.
T (if a transaction file is used)
(your program *must* use at least *one* file, either transaction or master if you are using WSU for data entry)
M (if master files are used)
. (one for each master file)

S
D
.
. (for first display)
C
.
.

S (one set of S-, D-, C-specifications
D for each additional display)
.
.
C
.
.

C (subroutine(s), if any)
.
.

SEU offers these specification display types to select from:



```
                          Select display

  1 Z            17 WSU-J       33 FORTRAN     49
  2 Z-LOWER      18 WSU-E       34 COBOL       50
  3 H            19 WSU-T       35 SDAS        51
  4 U            20 WSU-M       36 SDAH        52
  5 F            21 WSU-S       37 SDAD1       53
  6 G            22 WSU-D       38 SDAD2       54
  7 E            23 WSU-C       39 DOC-H       55
  8 L            24 SFGR-S      40 AUTOR       56
  9 T            25 SFGR-H      41 MESSAGE     57
 10 I            26 SFGR-D      42 SRT-HEAD    58
 11 J            27 D-CONT      43 SRT-RECD    59
 12 C            28 SORTH       44 SRT-CNST    60
 13 O            29 SORTRF      45 SRT-FLD     61
 14 P            30 SORTRC      46 MICRSYS     62
 15 K            31 SORTF       47 MICRSTCK    63
 16 A            32 ASSEM       48             64



 Enter the number of the specification display you want. . . . .
```

You use display types:

- WSU-J (number 17) to enter the J-specification
- WSU-E (number 18) to enter E-specifications
- WSU-T (number 19) to enter the T-specification, and
- WSU-M (number 20) to enter M-specifications.

To enter the S-specification(s) for a WSU program, you can use either display type WSU-S (number 21) or display type SDAS (number 35). You CANNOT use display type SFGR-S, because SEU does not allow you to enter data into WSU-only fields from the SFGR-S specification display.

To enter the D-specifications for a WSU program, you have several display choices:

Display type WSU-D (number 22) or
Display type SDAD1 (number 37).

If a constant needs to be continued on another statement, you can use either:

Display type D-CONT (number 27) or
Display type SDAD2 (number 38).

You *cannot* use display type SFGR-D to enter D-specifications for a WSU program, because SEU does not allow you to enter data into WSU-only fields from the SFGR-D specification display.

To enter C-specifications, you use display type WSU-C (number 23).

Storing the Source Program

The source program must be in a separate source member from the file definition.

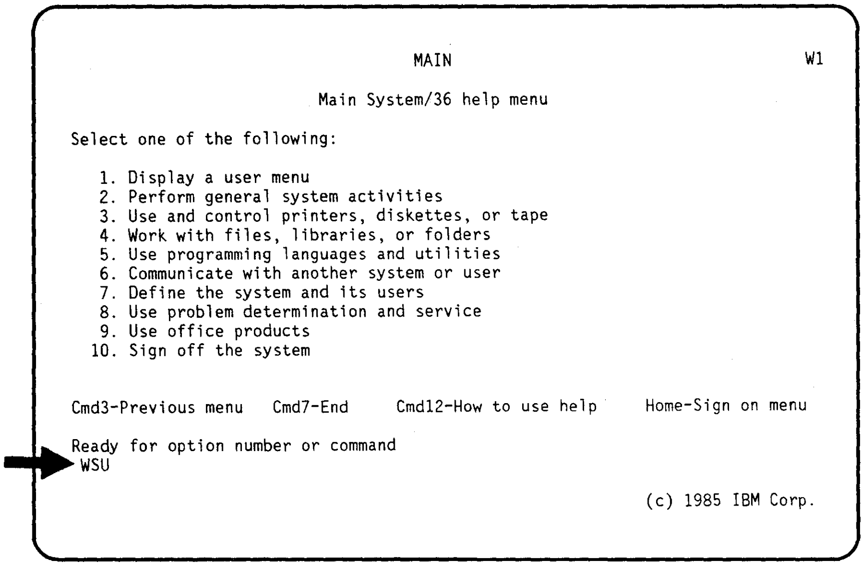
Generating a WSU Program

To generate your WSU program, you enter the WSU procedure command: either in its short form, to get the Work Station Utility prompt display, or in its long form, if you want to bypass the prompts.

The information you specify is the same whether you use the prompt display or the command format, but the options of the prompt display are in a slightly different sequence from the parameters of the command format.

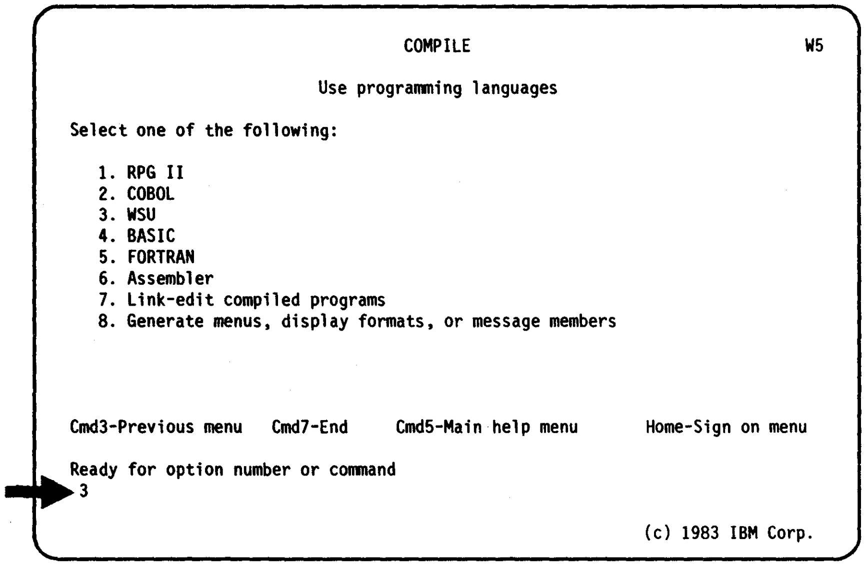
Using the WSU Prompt Display

To use the prompt display, enter WSU or HELP WSU on the command line. Or you can type WSU on the command line and press the Help key:



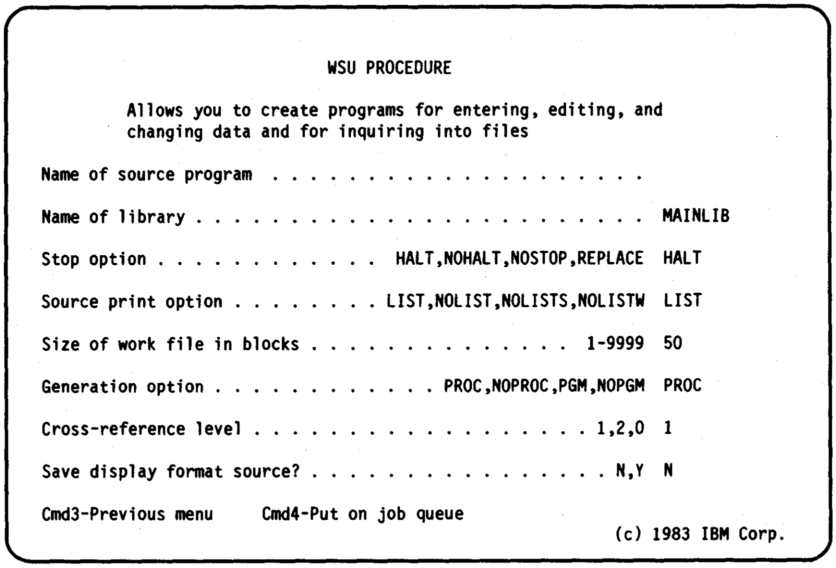
```
                                    MAIN                                     W1

                          Main System/36 help menu

   Select one of the following:

      1. Display a user menu
      2. Perform general system activities
      3. Use and control printers, diskettes, or tape
      4. Work with files, libraries, or folders
      5. Use programming languages and utilities
      6. Communicate with another system or user
      7. Define the system and its users
      8. Use problem determination and service
      9. Use office products
     10. Sign off the system


   Cmd3-Previous menu   Cmd7-End     Cmd12-How to use help    Home-Sign on menu

   Ready for option number or command
    WSU
                                                           (c) 1985 IBM Corp.
```

or you can choose option 3 of the COMPILE menu:



```
                                   COMPILE                                   W5

                          Use programming languages

   Select one of the following:

      1. RPG II
      2. COBOL
      3. WSU
      4. BASIC
      5. FORTRAN
      6. Assembler
      7. Link-edit compiled programs
      8. Generate menus, display formats, or message members



   Cmd3-Previous menu   Cmd7-End     Cmd5-Main help menu      Home-Sign on menu

   Ready for option number or command
    3
                                                           (c) 1983 IBM Corp.
```

The following display appears, prompting for the information WSU needs to generate your program:



```
                              WSU PROCEDURE

      Allows you to create programs for entering, editing, and
      changing data and for inquiring into files

Name of source program . . . . . . . . . . . . . . . . . . . .

Name of library . . . . . . . . . . . . . . . . . . . . . . . .  MAINLIB

Stop option . . . . . . . . . . . .  HALT,NOHALT,NOSTOP,REPLACE  HALT

Source print option . . . . . . . . LIST,NOLIST,NOLISTS,NOLISTW  LIST

Size of work file in blocks . . . . . . . . . . . . . . 1-9999  50

Generation option . . . . . . . . . . . . PROC,NOPROC,PGM,NOPGM  PROC

Cross-reference level . . . . . . . . . . . . . . . . . . 1,2,0  1

Save display format source? . . . . . . . . . . . . . . . . N,Y  N

Cmd3-Previous menu     Cmd4-Put on job queue
                                                        (c) 1983 IBM Corp.
```

Default values, representing the most common choices, are shown for some of the options (or parameters); you can either take the default values or change them. The following paragraphs describe the parameters and the values you can choose for them.

Name of Source Program

The name of the member that contains your source program is a required parameter. If you leave a blank here, WSU prompts for the source program name by asking you for the missing required parameter.

Name of Library

This parameter tells WSU the name of the library that contains your source program. If you do not specify the library name, WSU assumes the program is in the current library. If the source program is not found in the named (or assumed) library, WSU issues an error message.

Stop Option

This parameter specifies how processing should proceed during program generation. The choices are:

HALT	If a terminal error occurs, WSU stops processing and issues a message to the operator. If any duplicate library members are found, WSU issues a message and allows the operator to either replace the existing member or cancel the WSU procedure. HALT is the default.
NOHALT	If a terminal error occurs, WSU neither stops processing nor issues a message to the operator.
NOSTOP	A combination of NOHALT and REPLACE: If a terminal error occurs, WSU does not stop processing, but if members with duplicate names exist, WSU automatically replaces them with any newly generated members.
REPLACE	If any members existing in the library have the same name as any program, procedure, or format members to be generated, WSU automatically replaces the existing members with the members created during this generation.

If neither REPLACE nor NOSTOP is specified, a message appears for the operator each time a member with a duplicate name is found.

Source Print Option

This parameter specifies how much information you want WSU to include in the printout produced when the program is generated. The choices are:

LIST	Print a complete WSU program generation printout, including: – heading information. – source information. – diagnostic information. – display format information. LIST is the default value.
NOLIST	Print only diagnostic information.
NOLISTS	Print heading, source, and diagnostic information, but omit display format information.
NOLISTW	Print only diagnostic information and display format information.

Note: The actual contents of the printout also depend on the values that are specified for the *Generation Option* and *Cross-Reference Level* parameters, as well as any errors that occur during program generation.

An example of a generation printout is shown in Chapter 9, *Testing and Debugging WSU Programs*.

Size of Work File in Blocks

This parameter indicates the number of blocks that WSU should allocate for a work file. You can choose any number from 1 through 9999. Unless you specify otherwise, WSU allocates the default, 50 blocks.

Generation Option

This parameter specifies what generation output (besides the printout) you want WSU to produce. The choices are:

PROC	Generate a procedure that calls the WSU program (as well as the subroutine program and display formats). PROC is the default value.
NOPROC	Generate only the program and the display formats; do not generate a procedure.
PGM	Generate only the WSU program; do not generate a procedure or the display formats.
NOPGM	Generate only the printout; do not generate a program, a procedure, or display formats.

Note: The actual contents of the printout also depend on the values that are specified for the *Source Print Option* and *Cross-Reference Level* parameters, as well as any errors that occur during program generation.

An example of a generation printout is shown in Chapter 9, *Testing and Debugging WSU Programs*.

Cross-Reference Level

This parameter describes the extent of cross-reference information you want WSU to produce in the generation printout. The default is 1; the choices are:

0 Print the following:

- Heading information
- Source information
- Extended diagnostics
- Undefined indicators
- Field names defined more than once
- Undefined field names
- Main storage requirements
- Disk storage requirements
- Procedure generated for WSU program execution
- Diagnostic text.

1 Print the following:

- All information included for option 0
- Indicators used
- Unreferenced indicators
- User message member codes used
- Mode-level data field names used
- Session-level data field names used
- Job-level data field names used
- Program label names used
- Unreferenced field names.

2 Print the following:

- All information included for option 0
- All information included for option 1
- Indicator name usage cross-reference
- Field name and label usages cross-reference.

Note: The actual contents of the printout also depend on the values that are specified for the *Source Print Option* and *Generation Option* parameters, as well as any errors that occur during program generation.

An example of a generation printout is shown in Chapter 9, *Testing and Debugging WSU Programs.*

Save Display Format Source

If you plan to use help displays (which cannot be generated using WSU), you will want to specify a Y for this option. Refer to the manual, *Creating Displays*, for information on how to use help text. The default is N.

Cmd 4-Put on JOBQ

If your job does not require input from the operator while the job is running, the job can be placed on the job queue. When you place a job on the job queue, you can continue to use your display station for other work instead of waiting for your job to complete. You would normally place long-running jobs and jobs that do not need to run immediately on the job queue.

WSU Procedure Command

You can bypass the prompt display and enter the full WSU procedure command directly in the following format:

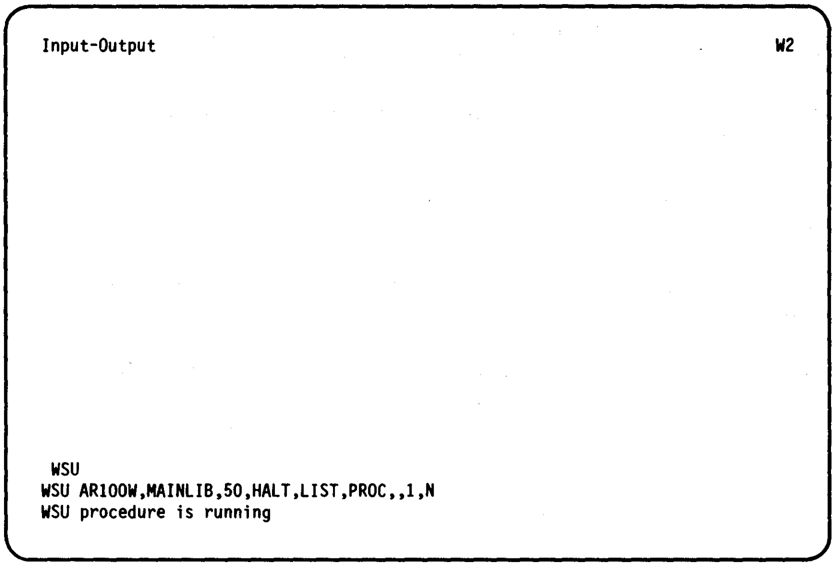
```
WSU      source member name, |source member library|, |block size|, |HALT   | ,
                             |current library      |  |50        |  |NOHALT |
                                                                     |NOSTOP |
                                                                     |REPLACE|

         |LIST    |, |PROC   |, |N|, |0|
         |NOLIST  |  |NOPROC |  |Y|  |1|
         |NOLISTS |  |PGM    |       |2|
         |NOLISTW |  |NOPGM  |
```

The parameters must be specified in the order shown. The order for the procedure command is:

- WSU
- Source member name
- Source library
- Block size
- Stop option
- Source print option
- Generation option
- Job queue option
- Cross-reference level.

The WSU Generation Process

When the WSU command is entered, either using the WSU prompt display or from the command display, the system displays a message:



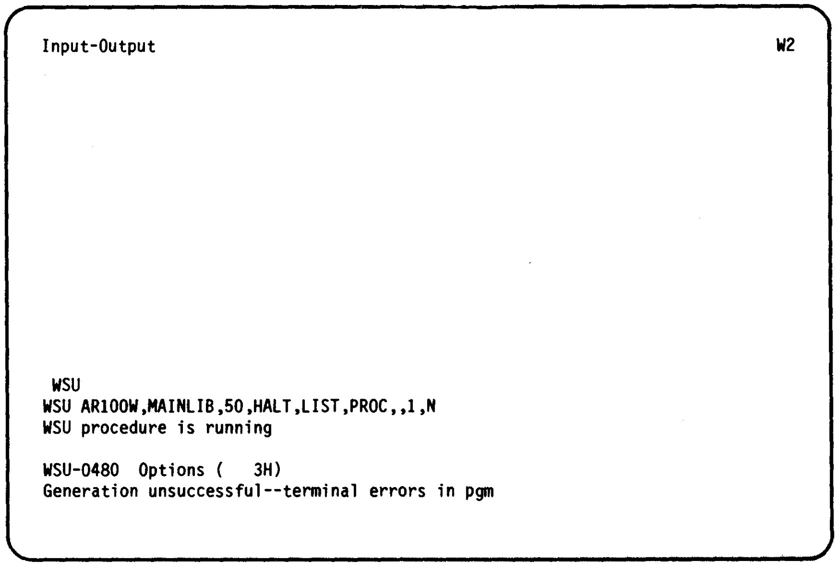
```
Input-Output                                                W2












 WSU
WSU AR100W,MAINLIB,50,HALT,LIST,PROC,,1,N
WSU procedure is running
```

After a slight wait, either:

- The printer starts your generation printout and your menu is displayed again (which means everything went right)

  or

- WSU displays another message:



```
Input-Output                                                W2












 WSU
WSU AR100W,MAINLIB,50,HALT,LIST,PROC,,1,N
WSU procedure is running

WSU-0480  Options (   3H)
Generation unsuccessful--terminal errors in pgm
```

(which means something went wrong).

When you respond to this message with the required 3-option, WSU produces an error printout. Chapter 9 describes the error printout and how to use it to debug your program.

Generation Output

When your program is successfully generated, WSU normally creates the following items (in addition to the generation printout) and stores them in the library you specified on the J-specification:

- A display format load member (which has the display *format member name* you gave on the J-specification)
- A procedure member (which has the *program name* you gave on the J-specification)
- A subroutine member (which has the same *program name* as the procedure).

The generated printout is described in detail in Chapter 9, *Testing and Debugging WSU Programs*.

The generated procedure member and ways in which you can modify it are described in the following section.

WSU-Generated Procedure

The procedure that WSU generates to call a program is an MRT (multiple requester terminal) type of procedure. There may be times that you will want to modify this procedure. If so, refer to the *Concepts and Programmer's Guide* for information on how to modify an MRT procedure.

Following is an example of a procedure generated for a WSU program:

```
// ATTR NEP-NO,MRTMAX-Ⓐ
// REGION SIZE-Ⓑ
// LOAD #WSXI1
// FILE NAME-Ⓒ
// IF DATAF1-Ⓒ ,DISP-OLD
// ELSE DISP-NEW,RECORDS-Ⓓ
// FILE NAME-Ⓔ ,DISP-OLD
// RUN
// WSX OBJLIBR-Ⓕ ,OBJMBR-Ⓖ ,MSGLIBR-Ⓗ ,MSGMBR-Ⓘ,
//  FMTLIBR-Ⓙ ,FMTMBR- Ⓚ ,DEBUG- Ⓛ ,UNIV-Ⓜ  ,
// END
```

where:

Ⓐ is the maximum number of display stations (from the J-specification)

Ⓑ is the region size required to run the program (from the J-specification)

Ⓒ is the transaction file name (from the T-specification)

Ⓓ is the number of records for the transaction file (from the T-specification)

Ⓔ is the master file name (from the M-specification)

Ⓕ is the name of the library that contains the WSU program (from the J-specification) [optional]

Ⓖ is the WSU program name (from the J-specification)

Ⓗ is the name of the library that contains your message member (from the J-specification) [optional]

Ⓘ is the name of the load member that contains your messages (from the J-specification) [optional]

Ⓙ is the name of the library that contains the display formats load member (from the J-specification) [optional]

Ⓚ is the name of the load member that contains the display formats (from the J-specification)

Ⓛ is the debugging option [optional]

Ⓜ is the universal work session selection authorization [optional]

Modifying the WSU-Generated Procedure

You can modify the generated procedure using SEU. In response to the SEU procedure prompt, you must enter:

- The name of the procedure member to be updated (the name is the same as that of the program).
- The type of member it is to be: P (procedure).
- The name of the library that contains the member (the name you specified on the J-specification).

You use SEU specification display type Z (number 1) for procedure members.

Some possible changes that you might want to make are described in the following paragraphs.

Maximum Number of Display Stations

To change (without modifying the source program) the maximum number of display stations and thus the number of operators who can use the program at the same time, you can change the MRTMAX parameter value in the ATTR statement to the new number of display stations.

The MRTMAX value determines the amount of main storage WSU allocates for the display stations. The amount of storage varies from application to application; however, WSU allocates from 25 bytes to 620 bytes of storage per display station.

Changing the MRTMAX value reduces or increases the required region size, which could determine whether or not a program runs in a region. For example, if MRTMAX is 8 but only one display station operates, you can save 155 to 4340 bytes of main storage by reducing MRTMAX to 1. Also, the size of the WSU work file increases as MRTMAX increases. WSU prints, on the generation printout, the amount of storage required for each work session.

```
// ATTR NEP-NO,MRTMAX-
```

Message Member Name and Library

If your program issues user messages (other than those from a MSG or IMSG operation) but you did not specify a message member on the J-specification when you generated your program (or if the messages are now in a different message member), you can modify the procedure to tell WSU where to find the user messages. Before calling the program, you can add MSGLIBR and MSGMBR parameters to the WSX statement in the order shown.

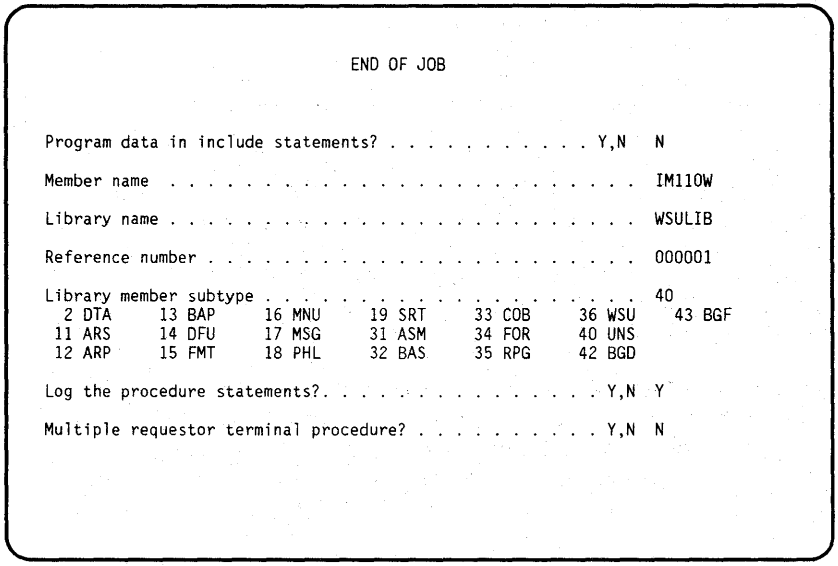
```
// WSX OBJLIBR- ,OBJMBR- , MSGLIBR-[          ],
    MSGMBR-[     ],
```

History File Logging

To reduce WSU startup time, you can change the generated procedure so that all statements are not logged to the history file. You make this change by giving a negative (N) response to the prompt:

```
Log the procedure statements?
```

on the second SEU End-of-Job display:



```
                             END OF JOB


Program data in include statements? . . . . . . . . . . . Y,N   N

Member name . . . . . . . . . . . . . . . . . . . . . . . . .   IM110W

Library name . . . . . . . . . . . . . . . . . . . . . . . . .  WSULIB

Reference number . . . . . . . . . . . . . . . . . . . . . . .  000001

Library member subtype . . . . . . . . . . . . . . . . . . . .  40
  2 DTA      13 BAP     16 MNU     19 SRT     33 COB     36 WSU    43 BGF
 11 ARS      14 DFU     17 MSG     31 ASM     34 FOR     40 UNS
 12 ARP      15 FMT     18 PHL     32 BAS     35 RPG     42 BGD

Log the procedure statements?. . . . . . . . . . . . . . . Y,N  Y

Multiple requestor terminal procedure? . . . . . . . . . . Y,N  N
```

Never-Ending Program

To make your program a never-ending program, specify NEP-YES on the ATTR statement. This change is best for WSU programs that are used most frequently, because it keeps the program active in storage even when no operators are using it. To end a never-ending program, you must set the EJ indicator on in the program or have the system operator stop the program.

```
// ATTR NEP-YES MRTMAX-N
```

Region Size

As the value for the REGION statement in the WSU-generated procedure, WSU uses the larger of:

- The region size that you code in columns 41 and 42 of the J-specification.
- The minimum region size.

Factors that affect ideal region size include:

- Required/accepted program performance.
- Region size limitations (maximum and minimum).
- Relative importance of the performance of the generated WSU program compared to other programs that run with the WSU program.

WSU prints, on the generation printout, the minimum region size and maximum region size that you can code for the generated program. WSU also prints the amount of storage required for each work session, which is useful information when MRTMAX is changed. The generation printout is described in detail in Chapter 9, *Testing and Debugging WSU Programs*.

You can vary the REGION SIZE parameter from one run to another to experiment with the effect the region size has on WSU performance. WSU adjusts to the region you specify, either taking advantage of additional space or running in less space (down to the minimum space specified on the output printed during generation).

```
// REGION SIZE-
```

File Disposition

You can change any file's disposition from OLD to SHR. When a master file is shared and updated, however, these operations should not occur between the GET operation and the file update: PUTS, MSG, IMSG, GET, and PUT.

Sharing the transaction file with an update program is not recommended because another operator may change a record before you are through.

```
// IF DATAF1- ,DISP-SHR
```

Debugging Option

The DEBUG-YES parameter of the WSX statement indicates that any debugging operations included on the program's C-specifications are to be done; DEBUG-NO (or no DEBUG parameter at all) indicates that debugging operations are to be ignored.

```
// FMTLIBR- ,FMTMBR- ,DEBUG-YES,UNIV-
```

Universal Work Session Selection Authorization

If display station security is not a concern for you, and if you want operators to be able to access data entered from more than one display station, you can use the UNIV parameter of the WSX control statement to allow such flexibility.

This authorization allows the operator to select a work session ID that is different from the symbolic ID of the display station being used.

- UNIV-ALL: Specifies that all display stations are authorized for such unrestricted session selection

- UNIV-id: Specifies the symbolic ID of the only display station from which unrestricted session selection can occur

  or

  Specifies a 2-character alphabetic authorization ID the operator can enter (on the WSU display EW line) to allow unrestricted session selection from the display station the operator is using.

```
// FMTLIBR-,FMTMBR- ,DEBUG- ,UNIV-ALL
```

Chapter 9. Testing and Debugging WSU Programs

This chapter includes some considerations and suggestions for *debugging* (finding and correcting errors in) your WSU program. Debugging is a straight forward process:

1. Is there an error?
2. What type of error is it?
    a. Error in generating?
    b. Error in running?
3. Correct the error.
4. Retry.

The types of errors that you might find in your program could be:

- Misinterpretation of the problem statement
- Faulty logic
- Incomplete coding of specifications
- Incorrect entry of specifications.

If the errors appear when you are trying to *generate* the program, they are probably *violations of syntax*; that is, you probably coded the specifications incorrectly or typed them incorrectly when you were using SEU to enter them.

If the errors appear when you are trying to *run* the program, the problem is more likely your logic or your interpretation of the problem statement.

You will probably need some or all of the following aids to carry out your debugging successfully:

- Diagnostic checklist
- Supporting documentation
- Diagnostic tools
    - Debugging template
    - Reference manual
    - Generation printout.
- Sample test data
    - Different record types
    - Undefined records
    - Maximum values
    - Error situations.

Diagnostic Checklist

Following is a possible checklist to follow when debugging your WSU program:

For Errors in Generation

- __ Were there any terminal (T) errors when you tried to generate the program?
  - __ Were the errors in text (obvious syntax errors on the specifications)?
  - __ Were the errors in defining fields/indicators?
  - __ Or something else?
    - __ Read the error message
    - __ Find the statement(s) in error
    - __ Read the error message explanation (in *Utilities Messages* manual)
    - __ Correct the coding and use SEU to change the source member
    - __ Try to generate the program again.

  (You can follow the same procedure for warning (W) messages produced in the generation, if you want to fix them.)

For Errors in Running the Program

When you attempted to run the program with test data,

- __ Did the first display appear?
  - __ Did it look like it was supposed to?
  - __ Did it accept (or reject) test data properly?
  - __ Did the command keys (WSU and user) work as they are supposed to?
- __ Did the next display appear when it was supposed to?
  - __ Did it look like it was supposed to?
  - __ Did it accept (or reject) test data properly?
  - __ Did the command keys (WSU and user) work as they are supposed to?
- __ Did the program work properly?
- __ Did the program end properly?
- __ Did the transaction file get entered/changed correctly?
- __ Did the master file get changed correctly?

Supporting Documentation

Supporting documentation would include your descriptions of the job to be done (like record format sheets, display format sheets, etc.).

Diagnostic Tools

Debugging Template and Reference Manual

The *WSU/$SFGR Debugging Template*, GX21-7926, includes column headings to match the RPG F- and I-specifications; the S- and D-specifications; and the WSU J-, E-, T-, M-, and C-specifications (as well as some other reference information). The spacing on the template is the same as the spacing on the generated printout, so you can determine what entries you have (or don't have) in the various columns of your file definition and source program specifications.

The main reference manual you will need is this manual, the *WSU Guide*; it contains reference information for completing specifications and program examples. Descriptions of all the messages that WSU issues (those printed at generation as well as those displayed when the programs are run) can be found in the *Utilities Messages* manual. You might also need the *System Messages* manual if you get messages from the system (labeled SYS-nnnn) as well as from WSU.

Generation Printout

The printout produced when the program is generated (or not generated!) can include heading information, source information, diagnostic information, and display format information.

The actual contents of the listing depend on the values specified for the *Source Print Option* parameter, the *Generation Option* parameter, the *Cross-Reference Level* parameter, and the types of program generation errors that occur. The examples in the following paragraphs are marked to show which parts of the printout result from each of the cross-reference level options.

Heading Information

Ⓐ Release level of WSU

Ⓑ Date of program generation

Ⓒ Time of program generation

Ⓓ Name of the library that contains the WSU source program

Ⓔ WSU program name (the name of the source member that contains the WSU source program)

Ⓕ Page number of WSU printout

```
IBM SYSTEM/36 WSU RELEASE 03                                          ⒷDATE  12/08/85  ⒸTIME  10.49
                   Ⓐ
                                                                       ⒹBFEBAD   IM11WBⒺ    PAGE   0001Ⓕ
```

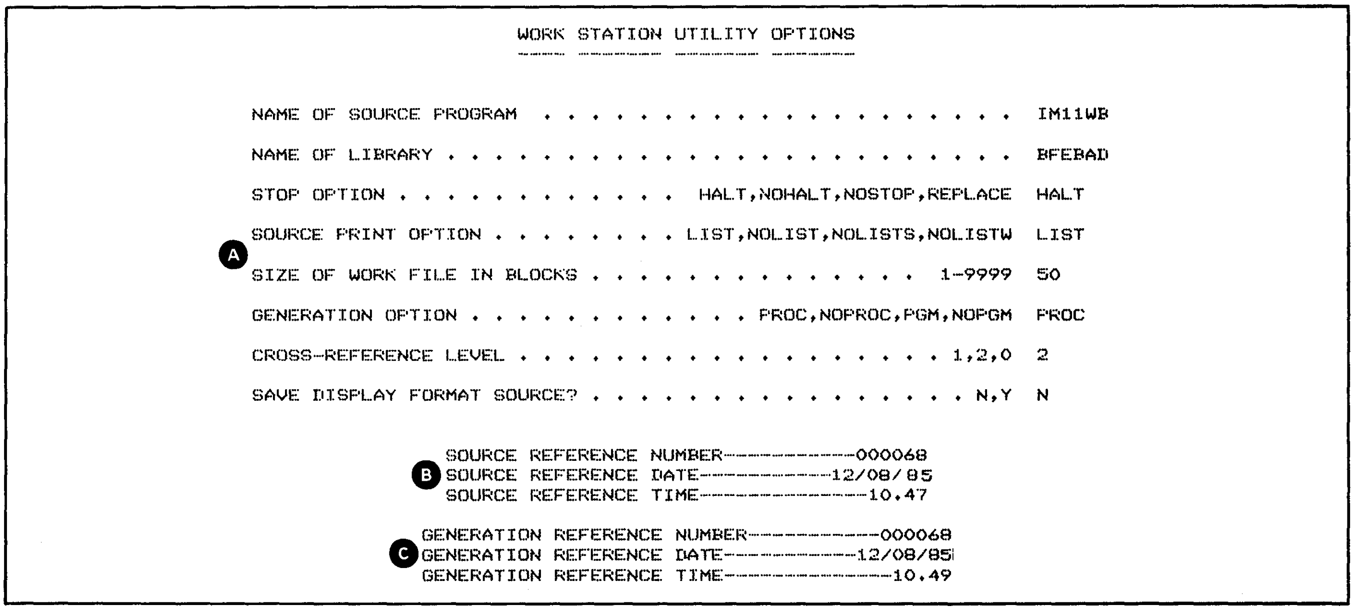
Work Station Utility Options

In addition to the heading information, the front page of your generated printout contains:

Ⓐ Options and defaults taken on the WSU procedure prompt

Ⓑ Number, date, and time of the latest changes to the WSU source program

Ⓒ Number, date, and time of the latest generation of this WSU program



```
                    WORK STATION UTILITY OPTIONS
                    ---- ------- ------- -------

NAME OF SOURCE PROGRAM  . . . . . . . . . . . . . . . . . . . .  IM11WB

NAME OF LIBRARY . . . . . . . . . . . . . . . . . . . . . . . .  BFEBAD

STOP OPTION . . . . . . . . . . . .  HALT,NOHALT,NOSTOP,REPLACE  HALT

SOURCE PRINT OPTION . . . . . . . . LIST,NOLIST,NOLISTS,NOLISTW  LIST

SIZE OF WORK FILE IN BLOCKS . . . . . . . . . . . . . .  1-9999  50

GENERATION OPTION . . . . . . . . . . . PROC,NOPROC,PGM,NOPGM  PROC

CROSS-REFERENCE LEVEL . . . . . . . . . . . . . . . . . . 1,2,0  2

SAVE DISPLAY FORMAT SOURCE? . . . . . . . . . . . . . . . . N,Y  N

         SOURCE REFERENCE NUMBER-------------000068
         SOURCE REFERENCE DATE-------------12/08/ 85
         SOURCE REFERENCE TIME---------------10.47

       GENERATION REFERENCE NUMBER-----------000068
       GENERATION REFERENCE DATE-----------12/08/85
       GENERATION REFERENCE TIME-------------10.49
```

The above printout is from a program using the WSU prompt display. If you had used the WSU procedure command, any defaults that you used would not appear on the generated printout.

Source Program Information

Ⓐ WSU source statements and file definition

J-specification

E-specifications

T-, F-, and I-specifications for the transaction file

M-, F-, and I-specifications for each master file

M- and I-specifications for the local data area fields

M- and I-specifications for session level fields

S-specification for each display

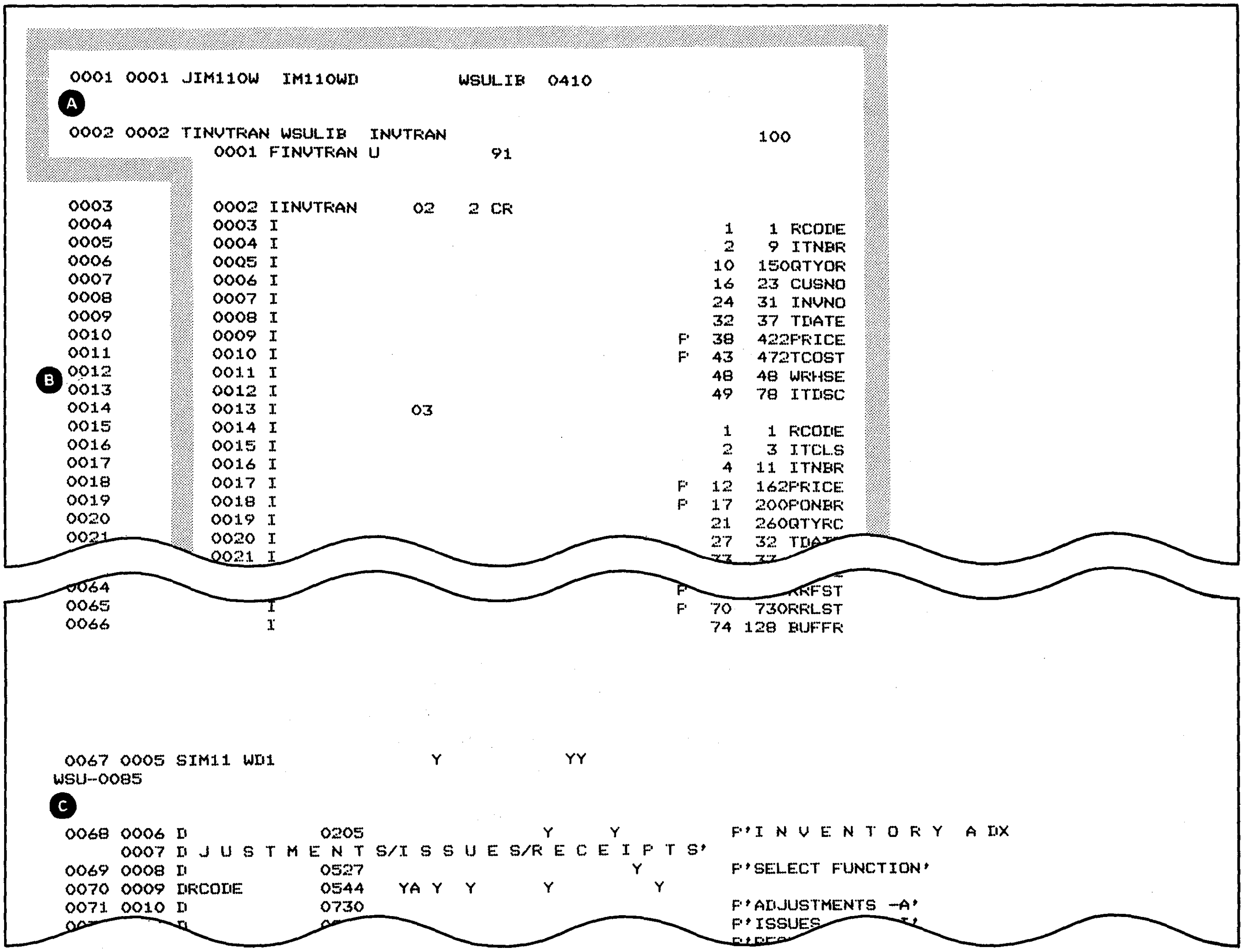
D-specifications for each display

C-specifications for each display

C-specifications for each subroutine

Ⓑ WSU prints a 4-digit statement number in front of each source statement. These numbers are referenced in the diagnostic information that follows the source statements. Comment lines, continued lines, F-specifications, and unrecognized or out-of-sequence specifications do not have statement numbers.

Ⓒ WSU error message numbers print beneath those source statements that contain errors.



```
 0001 0001 JIM110W   IM110WD          WSULIB  0410

 0002 0002 TINVTRAN WSULIB  INVTRAN                               100
                0001 FINVTRAN U           91

 0003           0002 IINVTRAN      02    2 CR
 0004           0003 I                                        1    1 RCODE
 0005           0004 I                                        2    9 ITNBR
 0006           0005 I                                       10   150QTYOR
 0007           0006 I                                       16   23 CUSNO
 0008           0007 I                                       24   31 INVNO
 0009           0008 I                                       32   37 TDATE
 0010           0009 I                                     P 38   422PRICE
 0011           0010 I                                     P 43   472TCOST
 0012           0011 I                                       48   48 WRHSE
 0013           0012 I                                       49   78 ITDSC
 0014           0013 I            03
 0015           0014 I                                        1    1 RCODE
 0016           0015 I                                        2    3 ITCLS
 0017           0016 I                                        4   11 ITNBR
 0018           0017 I                                     P 12   162PRICE
 0019           0018 I                                     P 17   200PONBR
 0020           0019 I                                       21   260QTYRC
 0021           0020 I                                       27   32 TDA
                0021 I
 0064                                                      P         RRFST
 0065                I                                     P 70   730RRLST
 0066                I                                       74  128 BUFFR




 0067 0005 SIM11 WD1                Y              YY
WSU-0085

 0068 0006 D              0205                 Y     Y        P'I N V E N T O R Y  A DX
      0007 D J U S T M E N T S/I S S U E S/R E C E I P T S'
 0069 0008 D              0527                        Y       P'SELECT FUNCTION'
 0070 0009 DRCODE         0544   YA Y  Y       Y         Y
 0071 0010 D              0730                                P'ADJUSTMENTS -A'
                                                              P'ISSUES
```

Diagnostic Information

While all possible diagnostic information is described here, the actual information that is printed depends on the parameters specified on the WSU command and the types of errors that occur during program generation.

- Extended diagnostics: A cross-reference list of WSU message numbers and the statement number at which each error occurred.

```
   EXTENDED DIAGNOSTICS
   NOTE#           STMT#
WSU-0281            0082
WSU-0281            0090
WSU-0281            0115
WSU-0281            0146
WSU-0281            0173
```

- Indicator name usage: An alphabetic cross-reference list of indicators used in the program and the statement number in which each indicator appears.
  - An asterisk to the *left* of a statement number means that the indicator has been defined in one of the following ways:
    - Is either set on or set off
    - Defines a record type in columns 19 and 20 of an I-specification
    - Defines a processing level in columns 7 and 8 of a C-specification.
  - An asterisk to the *right* of a statement number means that a record-identifying indicator has been referenced in one of the following ways:
    - Used in columns 54 and 55 for a PUT operation on a C-specification
    - Used in columns 48 through 53 (review mode record identifying indicator) or columns 54 through 59 (insert mode record identifying indicator) on an S-specification.

```
INDICATOR NAME USAGES                    RIGHT * = RECORD TYPE REFERENCE
  NAME        STMT#                      LEFT  * = DEFINITION
  EW          *0077  *0113  *0144  *0171
  KG           0077   0078   0113   0114   0144   0145   0171   0172
  KH           0115   0146   0173
  01          *0024  *0033  *0053  *0083   0084   0089   0090   0123*  0123
  02          *0003  *0085   0086   0089   0090   0152*  0152
  03          *0014  *0087   0088   0089   0090   0178*  0178
  72          *0119   0120   0121
  89          *0032  *0052  *0079  *0080   0081   0082
```

- Indicator lists provide a record of those indicators that are:
  - **Used**: Defined and referenced
  - **Unreferenced**: Defined but not referenced
  - **Undefined**: Referenced but not defined

```
INDICATORS USED
  EW KG KH 01 02 03 72 89
```

- User message member MICs used: A list of the MICs that are used in the program.
- Field name and label usages: An alphabetic cross-reference list of field names and labels that are used in the program and the statement number at which each name or label appears
  - An asterisk to the *left* of a statement number means that the field or label is defined at that statement
  - An asterisk to the *right* of a statement number indicates a possible update to a field, and means that the field is one of the following:
    - A result field on a C-specification
    - An input field on a D-specification
    - On a field-type I-specification for the transaction file
    - On a field-type I-specification for a master file that allows record additions.

```
FIELD NAME AND LABEL USAGES           RIGHT * = UPDATE REFERENCE
  NAME       STMT#                    LEFT  * = DEFINITION
  *ERROR
  *RLNO
  *RLRN
  *RLRR
  *RLRU
  *SLNO
  *USID
  *WSID
  ACODE      *0027*  0101*  0119
  ARCOD      *0034  *0054
  AVAIL      *0039
  BUFFR      *0051  *0066
  CUSNO      *0007*  0138*
  END         0078  *0091
  END1        0114  *0124
  END2        0145  *0153
  END3        0172  *0179
  INVNO      *0008*  0136*
  ITCLS      *0016* *0058
  ITDSC      *0013* *0023* *0031* *0056   0097   0130   0159
  ITNBR      *0005* *0017* *0026*  0032  *0035   0052  *0055   0075*  0095   0128   0157
  ITYPE      *0057
  MDATE      *0048  *0063
  ONORD      *0038
```

- Data field names used consist of alphabetic lists of all the *mode-level, session-level,* and *job-level* data fields used in the program. For each field, WSU lists the following information:

Column Heading	**Field Information**
NAME	Field name.
DIM	The number of elements in the array(s) in the program (blank in this example).
STMT#	The statement number in which the field is first defined, or RSVD if the field is a WSU reserved field.
LNG	The decimal length of the field. An asterisk to the right of the length indicates that WSU assumed an unpacked length for a packed field.
DEC	If the field is numeric, the number of decimal positions; if the field is alphameric, the letter A.
DISP	The hexadecimal displacement of the field into the data area.
LCL	The character U if the field is a local data area field.
F/ICNT	The number of times the field is referenced on all I-specifications (except session-level and local-data-area I-specifications).
SCR	The character D indicates an input or output field on a D-specification.
CHN	The character M indicates a GET field.

```
MODE LEVEL DATA FIELD NAMES USED
  NAME   DIM  STMT#   LNG    DEC    DISP  LCL F/ICNT SCR CHN
  RCODE       0004    001      A    0000       000    D
  ITNBR       0005    008      A    0001       000    D   M
  QTYOR       0006    006      0    0009       000
  CUSNO       0007    008      A    000F       000    D
  INVNO       0008    008      A    0017       000    D
  TDATE       0009    006      A    001F       000
  PRICE       0010    009      2    0025       000    D
  TCOST       0011    009      2    002E       000
  WRHSE       0012    001      A    0037       000    D
  ITDSC       0013    030      A    0038       000    D
  ITCLS       0016    002      A    0056       000
  PONBR       0019    007      0    0058       000    D
  QTYRC       0020    006      0    005F       000    D
  ACODE       0027    001      A    0065       000    D
  QTYIS       0028    006      0    0066       000    D
  UCOST       0061    009      2    006C       000    D
```

- For each program label name used, WSU lists:
  - The statement number where the label is initially defined
  - The operation (TAG, BEGSR, or ENDSR) that is used to define the label

```
PROGRAM LABEL NAMES USED
  NAME        STMT#     TYPE
  END         0091      TAG
  END1        0124      TAG
  END2        0153      TAG
  END3        0179      TAG
```

- Field names that are:
  - Unreferenced: Defined (on I- or C-specifications) but not used (on M-, D-, or C-specifications)
  - Defined more than once: Defined differently (on multiple I- or C-specifications)
  - Undefined: Referenced (on M-, D-, or C-specifications) but not defined I- or C-specifications)

```
WSU-0325 UNREFERENCED FIELD NAMES
         NAME       STMT#
         ARCOD      0034
         STOCK      0037
         ONORD      0038
         AVAIL      0039
         REORD      0040
         RENUM      0041
         QUSMO      0042
```

- A cross-reference list of displayed text for programmed halts and the statement number of the location each IMSG, MSG, or DEBUG literal appears

```
SOURCE/OBJECT CROSS REFERENCES
  STMT#     CSB  IAR  SSS     ****** DISPLAYED TEXT FOR PROGRAMMED HALTS (IMSG/MSG/DEBUG) ******
  0081      0000 0025 0009      ENTER VALID ITEM NUMBER
  0089      0000 0063 0009      INVALID FUNCTION
  0120      0007 003B 000A      INVALID CODE
```

- Storage requirements:
  - Main storage: For running the WSU program

```
MAIN STORAGE REQUIREMENTS FOR WSU PROGRAM EXECUTION          MINIMUM        MAXIMUM
 BYTES REQUIRED PER WORK STATION                                  28            284
 TIMES MAXIMUM NUMBER OF WORK STATIONS                             4              4
 PLUS ADDITIONAL STORAGE REQUIRED                               9824          47368
 YIELDS STORAGE REQUIRED FOR EXECUTION                          9936          48504

 REGION REQUIRED FOR EXECUTION                                   10K            48K
```

  - Disk storage: For the program work file

```
DISK STORAGE REQUIREMENTS FOR WSU EXECUTION PROGRAM WORKFILE
 SECTORS REQUIRED PER WORK STATION                                 12
 TIMES MAXIMUM NUMBER OF WORK STATIONS                              4
 PLUS ADDITIONAL DISK SECTORS REQUIRED                              5
 YIELDS MINIMUM DISK SECTORS REQUIRED                              53

 EXECUTION WORKFILE SIZE IN BLOCKS                                  6
```

  - The procedure generated for calling the execution program

```
IM110W   PROCEDURE CREATED FOR EXECUTION
 // ATTR NEP-NO,MRTMAX-4
 // REGION SIZE-10
 // LOAD #WSXI1
 // FILE NAME-INVTRAN,
 // IF DATAF1-INVTRAN DISP-OLD
 // ELSE DISP-NEW,RECORDS-100
 // FILE NAME-ITEMBALN,DISP-OLD
 // FILE NAME-ITEMMSTR,DISP-OLD
 // RUN
 // WSX OBJLIBR-WSULIB,OBJMBR-IM110W,FMTLIBR-WSULIB,FMTMBR-IM110WD
 // END
```

  - Diagnostic text: The message number, severity, and text for each program generation message

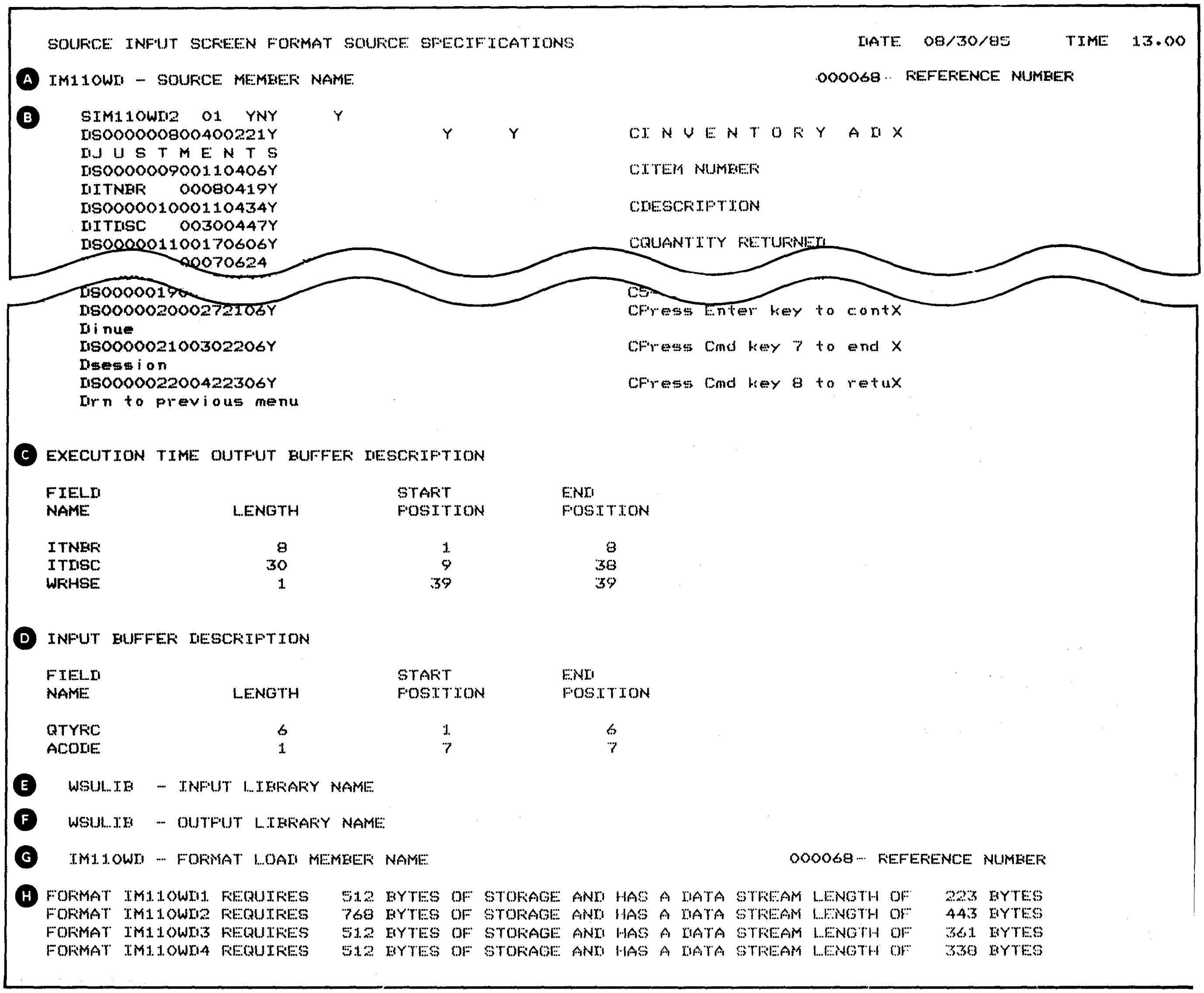
```
        DIAGNOSTIC TEXT
NOTE#    SEV         MESSAGE TEXT
WSU-0325  W  LISTED FIELD NAME DEFINED BUT NOT REFERENCED.
```

  - A message that indicates whether or not program generation was successful

```
                GENERATION SUCCESSFUL---WARNING ERRORS LISTED ABOVE.
```

Display Format Information

- **A** Name of the source member that contains the S- and D-specifications
- **B** S- and D-specifications generated for display format input
- **C** Execution time output buffer for the WSU program
- **D** Execution time input buffer for the WSU program
- **E** Input library name
- **F** Output library name
- **G** Format load member name
- **H** Amount of main storage required for each display screen format



```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS                          DATE  08/30/85     TIME  13.00
IM110WD - SOURCE MEMBER NAME                                            000068- REFERENCE NUMBER
   SIM110WD2  01  YNY     Y
   DS000000800400221Y              Y     Y          CI N V E N T O R Y  A D X
   DJ U S T M E N T S
   DS000000900110406Y                               CITEM NUMBER
   DITNBR   00080419Y
   DS000001000110434Y                               CDESCRIPTION
   DITDSC   00300447Y
   DS000001100170606Y                               CQUANTITY RETURNED
            00070624
   DS00000190                                       C5
   DS000002000272106Y                               CPress Enter key to contX
   Dinue
   DS000002100302206Y                               CPress Cmd key 7 to end X
   Dsession
   DS000002200422306Y                               CPress Cmd key 8 to retuX
   Drn to previous menu

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                         START          END
NAME           LENGTH         POSITION       POSITION

ITNBR            8              1              8
ITDSC           30              9             38
WRHSE            1             39             39

INPUT BUFFER DESCRIPTION

FIELD                         START          END
NAME           LENGTH         POSITION       POSITION

QTYRC            6              1              6
ACODE            1              7              7

  WSULIB  - INPUT LIBRARY NAME

  WSULIB  - OUTPUT LIBRARY NAME

  IM110WD - FORMAT LOAD MEMBER NAME                                   000068- REFERENCE NUMBER

FORMAT IM110WD1 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   223 BYTES
FORMAT IM110WD2 REQUIRES   768 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   443 BYTES
FORMAT IM110WD3 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   361 BYTES
FORMAT IM110WD4 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   338 BYTES
```

Testing

Program testing, the process of running a program with the intent of finding errors in it, is an important factor in the development of your application program. By testing your program, you can verify if your program:

- Is correctly accepting input data.
- Is correctly processing the input data.
- Is producing output correctly, whether the output is a file or a display format.
- Can handle unexpected and incorrect data.

You should be aware of the following considerations when you begin to test your programs.

Adequate Time for Testing: Allow plenty of time for testing your programs. If you have a complex program, you may need more time to test than you did to write the program.

Test Schedule and Testing Logbook: Documenting the testing process is important. You should have a written record of the level of testing that you did on your program. When you document your testing, be sure you have, for all cases, a description of:

- The test data used as input to the program
- The correct output you expected for the test data.

You should also assign some unique identification to each task in the testing process to aid you in the logging of the testing process.

Manageable Tasks: Divide the testing process into manageable tasks that proceed from the simple to the complex in terms of the functions performed in the program. Each task should build upon the functions of the preceding task. For example, if your program has to read or write three different record types, you should test to see if the program can handle one, two, and then all three record types. Generally, the first task you should do is to review your program code for errors in logic or syntax.

Appropriate Test Data

Test your programs with data that checks whether the program is correctly performing its intended functions. In most cases, your program cannot be tested in a single test because certain functions must be tested individually. You must also check the way these functions interconnect in your program. Label your test data for each test, and keep this data for future use in the event that you have to modify your program due to errors found in it during testing. Your test data should cover:

- Normal and expected data
- Error data and unexpected conditions.

Normal and Expected Data

When you test normal and expected conditions, use data that is representative of the real data that you will be using in the program. For example, in testing an order entry application, data could be used to test the following conditions:

- Opening a new account
- Updating an existing account
- Closing an existing account
- Updating multiple existing accounts.

Error Data and Unexpected Conditions

Use data in testing for errors and unexpected conditions that is not representative of the data that you would use in running your program. Data on error conditions is important because you may discover that your program is generating errors when used in a new or unexpected way. By using data that is erroneous or unexpected, you will have a way of seeing whether your program performs predictably.

Some examples of error condition testing are:

- Attempting transactions against nonexistent account numbers
- Updating closed accounts
- Using input data with invalid dates, incorrect totals, or invalid ranges of values in key fields
- Using combinations of data that have multiple errors or data that has a combination of valid and invalid values.

Some examples of unexpected condition testing are:

- Using no data as input to a program
- Running 2 days' worth of data as one day's data
- Running a program with the wrong input.

It is important when you are testing error conditions to ensure that your program is issuing error messages that describe the errors encountered in testing.

Other Testing Conditions

Check the following when you are testing a program:

- Have you successfully tested the restart or recovery procedures?
- Are the people who have to use the program familiar with the procedures needed to use or run the program?
- Does the program pass the right data to other programs that have to use the program's output?
- Are the system operators familiar with the requirements for the program?
- Has the program been tested for all phases of processing? For example, if the program has to generate weekly and monthly reports, have you merely tested for weekly report generation?

Program coding and testing should continue until all program functions are coded and tested. Using the DEBUG operation in complicated programs can shorten testing. If DEBUG is used, condition each DEBUG operation with an external indicator (for example, U8). Thus, the program can be tested with or without debugging without regenerating it. Another, perhaps easier, way to shorten your testing is to use DEBUG-NO. When using DEBUG-NO, you do not have to condition the external indicators.

As each program is being developed, changes may have to be made to the original specifications. For example, it may be easier to keep a field of information on the display rather than in the control record. This would mean changing the display and the logic of the program.

Chapter 10. ATTENTION: OPERATORS (Running a WSU Program)

The purpose of this chapter is to provide information about running WSU programs; therefore, it is intended primarily for operators (or programmers who are running their own WSU programs).

There are a few things about WSU that may be useful to know when you are running a WSU program, such as:

- Jobs
- Work sessions
- Operating modes
- Command/function keys
- WSU display and WSU aid display.

Those topics are discussed in the beginning of this chapter. Then, following those discussions, are descriptions of the tasks you might need to do while running a WSU application program:

- Starting a work session
- Entering data
- Reviewing records
- Inserting records
- Deleting records
- Ending a work session
- Restarting a work session.

JOBS

A WSU *job* is the entire time during which a WSU application program is run, from the time the first operator signs on to that program until the time the last operator signs off. Usually, more than one operator can use a WSU program at the same time.

Two or more different WSU jobs (for example, an order entry program and an inventory transactions program) can run at the same time.

WORK SESSIONS

A work session, to WSU, is the period from the first time an operator at a display station signs on to a job until an operator at that display station signs off the job.

Several display stations can be using the same job at the same time, so there can be as many work sessions as there are authorized display stations. If, for example, the maximum of four operators are authorized to use a program at one time, and you try to sign on when four operators are already using that program, you will not be able to sign on. You won't get a message saying that you can't sign on (or why), but you will have to either wait until one of the operators signs off or cancel your request, using the Inquiry display.

Any records entered from a particular display station are part of the corresponding *chain* of records for that work session in the transaction file. And, unless the programmer specifies otherwise, you can only review/insert/delete records in the chain of the work session (display station) you are using. So, for example, if you used a WSU program to enter records from display station X1, you wouldn't normally be able to review/update those records if you signed on to display station X2.

Operating Modes

You will see the word *mode* used quite a bit throughout this section. The modes we talk about (enter, review, insert, delete) are just the different kinds of work that a WSU program can be written to do. Because the program can't do everything at once, we say you are *in enter mode* when you are entering data, but *in review mode* when you ask the program to show you a previous record.

Some things to remember about modes:

- All programs are in enter mode when you start.
- You have to signal the program, using a command key, when you want to change from one mode to another.
- You can only change to review, insert, or delete mode if the program you are using was specifically written to allow you to review, insert, or delete records.

COMMAND KEYS AND FUNCTION KEYS

Figure 10-1 is an illustration of a display station keyboard. It shows the command keys (on the top row of the keyboard), the template that identifies them, and the function keys. The lowercase command keys are called Cmd 1 through Cmd 12; the uppercase keys are called Cmd 13 through Cmd 24.

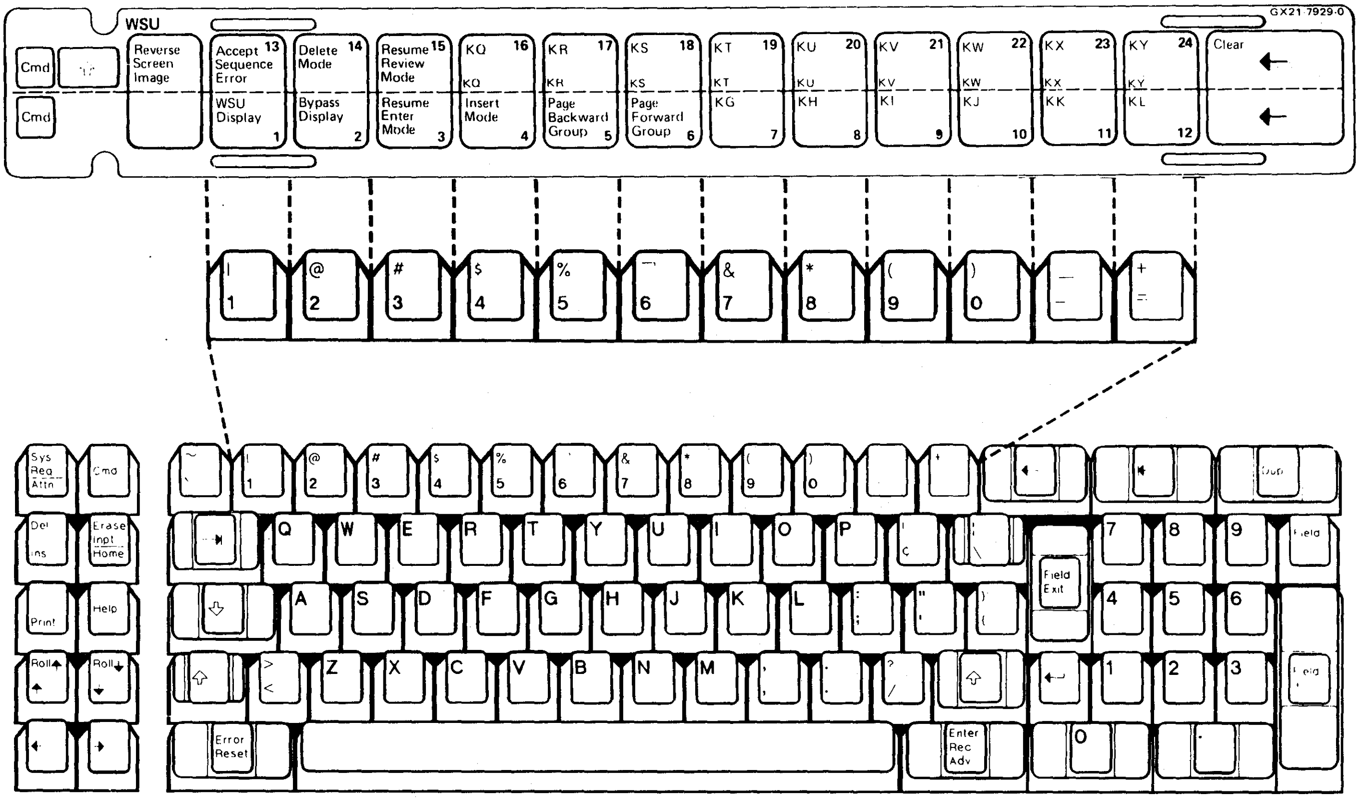


Figure 10-1. WSU Command/Function Keys and Template

WSU Command Keys

The Cmd 1 through 6 keys and Cmd 13 through 15 keys have special uses, and therefore special function names, when you are using WSU programs:

Name	**Key**	**Function Name**	**What It Does**
Cmd 1	| 1	WSU Display	Shows the WSU display.
Cmd 2	@ 2	Bypass Display	Shows the next appropriate display. You can use this command key to advance from a repeated display.
Cmd 3	# 3	Resume Enter Mode	In review, delete, or insert mode, changes the operating mode back to enter and shows again the entry display from which you selected another mode.
Cmd 4	$ 4	Insert Mode	In review mode, changes the operating mode to insert and shows a display for you to insert a record.
Cmd 5	% 5	Page Backward Group	Changes the operating mode to review and shows the previous header record in the transaction file.
Cmd 6	¬ 6	Page Forward Group	Changes the operating mode to review and shows the next header record in the transaction file.
Cmd 13	| 1	Accept Sequence Error	In enter mode (if the program is so written), allows you to bypass a required display. In review, delete, or insert mode, shows the WSU Aid display.
Cmd 14	@ 2	Delete Mode	Changes the operating mode to delete.
Cmd 15	# 3	Resume Review Mode	Changes the operating mode back to review.

User Command Keys

The Cmd 7 through 12 keys and Cmd 16 through 24 keys are called *user* command keys because the programmer can assign a use for them in the WSU program (they can also be disabled in the WSU program).

Pressing one of these keys:

- Enters the display. *Enters the display* refers to the operator pressing the Enter key, pressing a user-enabled command key, or leaving a field for which auto-record-advance has been specified.)
- Does whatever action the program has assigned to that key by setting on the associated indicator (KG, KH, . . .).

Function Keys

Two of the uppercase function keys have a special use with WSU programs:

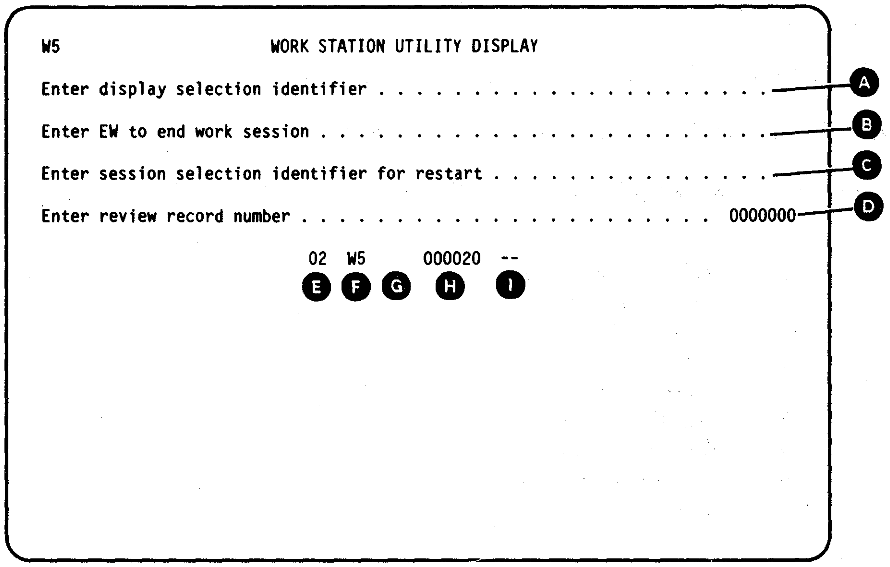
Key	**Function Name**	**What It Does**
Roll↓ ↓	Page Backward Record	Changes the operating mode to review (if it isn't already) and shows the previous record in the transaction file.
Roll↑ ↑	Page Forward Record	Changes the operating mode to review (if it isn't already) and shows the next record in the transaction file.

The rest of the function keys are used in their normal way.

The WSU Display

The WSU display appears when you press the WSU Display command key (Cmd 1). You can select this display anytime a display is waiting for input from you.

The first line of the display shows the display station identifier, the next record number, and the title of the menu. Figure 10-2 shows the WSU display, and the paragraphs following the illustration describe the information on the display.



```
W5                      WORK STATION UTILITY DISPLAY

Enter display selection identifier . . . . . . . . . . . . . . . . . . . . .

Enter EW to end work session . . . . . . . . . . . . . . . . . . . . . . . .

Enter session selection identifier for restart . . . . . . . . . . . . . . .

Enter review record number . . . . . . . . . . . . . . . . . . . . . . 0000000

                          02  W5     000020   --
```

Figure 10-2. The WSU Display

Ⓐ Enter display selection identifier

You can select a display by entering its 2-character ID on this line (but you can't select the ID of the display you just left).

Ⓑ Enter EW to end work session

You can end your work session by entering EW on this line.

If the program allows, you can select a work session ID that differs from the ID shown in the top left corner of the WSU display by entering the session selection authorization ID on this line.

Ⓒ Enter session selection identifier for restart

You can restart your work session or start a different work session by entering the appropriate work session ID on this line.

Ⓓ Enter review record number

You can review a specific record in the transaction file by entering the record's relative record number on this line.

Ⓔ Format/level ID field

The format ID of the current display (blank in the example)

The WSU processing level (IJ, IW, EW, or EJ)

** If there is no current processing

-- If current processing has been stopped by the operator before it is finished.

Ⓕ Session ID field

The session ID of the active session (W1 in the example).

Ⓖ Mode/processing level field (blank in the example)

A code identifying the current WSU mode or processing level:

Blank	Normal enter mode (as in the example)
RV	Review mode
IN	Insert mode
DL	Delete mode
IJ	Initiate job
IW	Initiate work session
ES	End sequence set
EW	End work session
EJ	End job

Ⓗ Add/update relative record number field

The current record number for add or update operations (9 in the example)

****** If there is no current review record during review mode

000000 If there is no transaction file.

Ⓘ Chain-end indicator field

-- If review mode is not supported in the program (as in the example)

Blank if review mode is supported but not active or if you are not at the end of the chain

** If you are at the end of the chain. If you press one of the Page Forward keys at end-of-chain, you will get the first record in the chain; if you press a Page Backward key, you will get the last record in the chain.

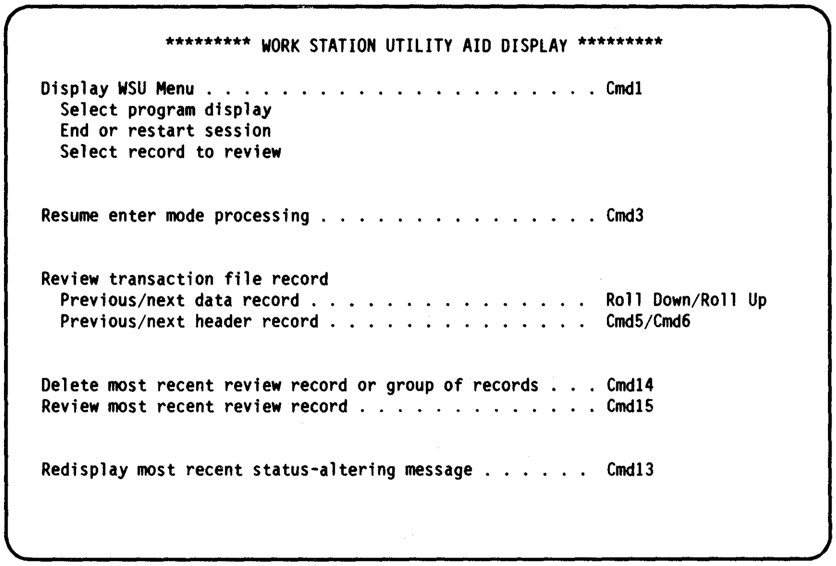
Returning from the WSU Display

If you displayed the WSU display while you were in enter mode or reviewing, deleting, or inserting a record, you can return to the display from which you requested the WSU display by pressing the:

- WSU Display command key (Cmd 1), in which case the entries on the display are ignored
- Enter key, in which case the entries on the display must all be blanks or zeros.

WSU Aid Display

When you are in review, insert, or delete mode, WSU provides a display that can help you request some functions:



```
             ********* WORK STATION UTILITY AID DISPLAY *********

Display WSU Menu . . . . . . . . . . . . . . . . . . . . . Cmd1
  Select program display
  End or restart session
  Select record to review


Resume enter mode processing . . . . . . . . . . . . . . . Cmd3


Review transaction file record
  Previous/next data record . . . . . . . . . . . . . . .  Roll Down/Roll Up
  Previous/next header record . . . . . . . . . . . . . .  Cmd5/Cmd6


Delete most recent review record or group of records . . . Cmd14
Review most recent review record . . . . . . . . . . . . . Cmd15


Redisplay most recent status-altering message . . . . . .  Cmd13
```

You can get this display by pressing the:

- Accept-Sequence-Error command key (Cmd 13) from a WSU or program-defined display.
- Enter key or a program-defined key when there is no current processing.
- WSU Display command key (Cmd 1) from the WSU display when there is no current processing.

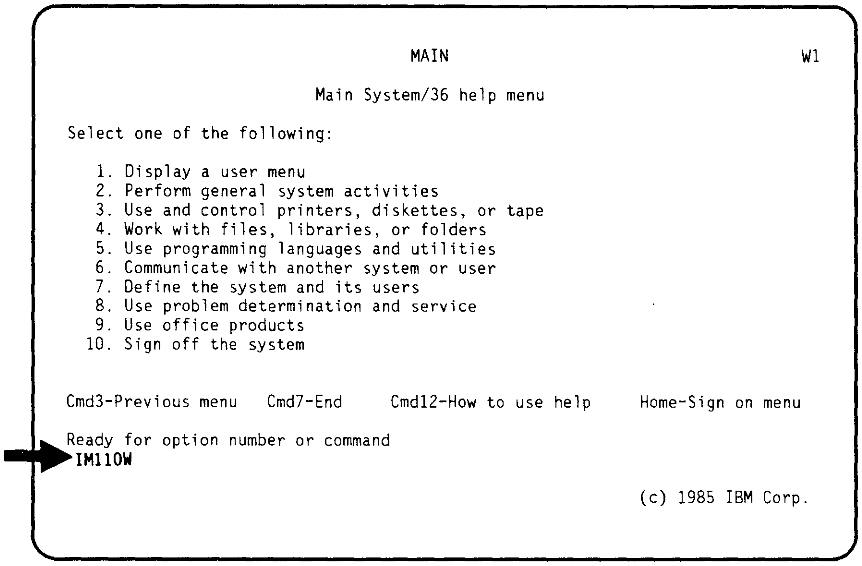
Starting a WSU Program

To use a WSU program you can:

- Enter its name in response to the

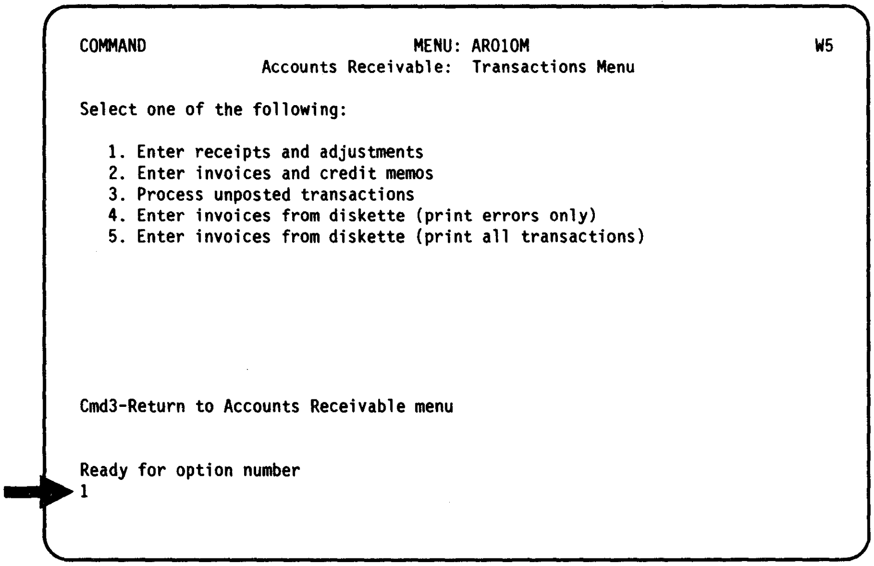
  `Ready for option number or command`

  prompt (in this example, the program name is IM110W):



```
                               MAIN                                  W1

                       Main System/36 help menu

Select one of the following:

   1. Display a user menu
   2. Perform general system activities
   3. Use and control printers, diskettes, or tape
   4. Work with files, libraries, or folders
   5. Use programming languages and utilities
   6. Communicate with another system or user
   7. Define the system and its users
   8. Use problem determination and service
   9. Use office products
  10. Sign off the system

Cmd3-Previous menu   Cmd7-End     Cmd12-How to use help     Home-Sign on menu

Ready for option number or command
IM110W
                                                     (c) 1985 IBM Corp.
```

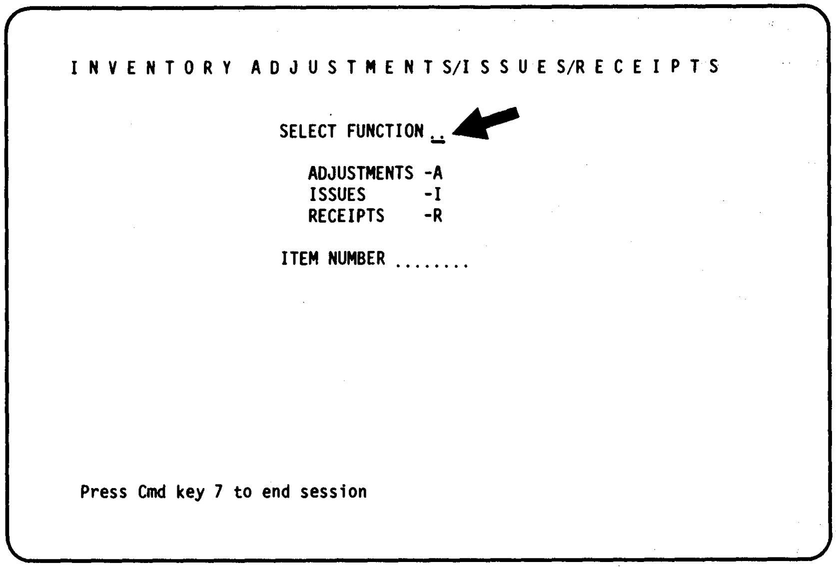
- Select the appropriate number from a menu that the programmer has provided (in this example, the program is option 1):



```
COMMAND                      MENU: AR010M                            W5
                  Accounts Receivable:  Transactions Menu

Select one of the following:

   1. Enter receipts and adjustments
   2. Enter invoices and credit memos
   3. Process unposted transactions
   4. Enter invoices from diskette (print errors only)
   5. Enter invoices from diskette (print all transactions)




Cmd3-Return to Accounts Receivable menu


Ready for option number
1
```

Entering Data on a Display

When a display appears, the program places the cursor at the first input field, as shown on the following display:



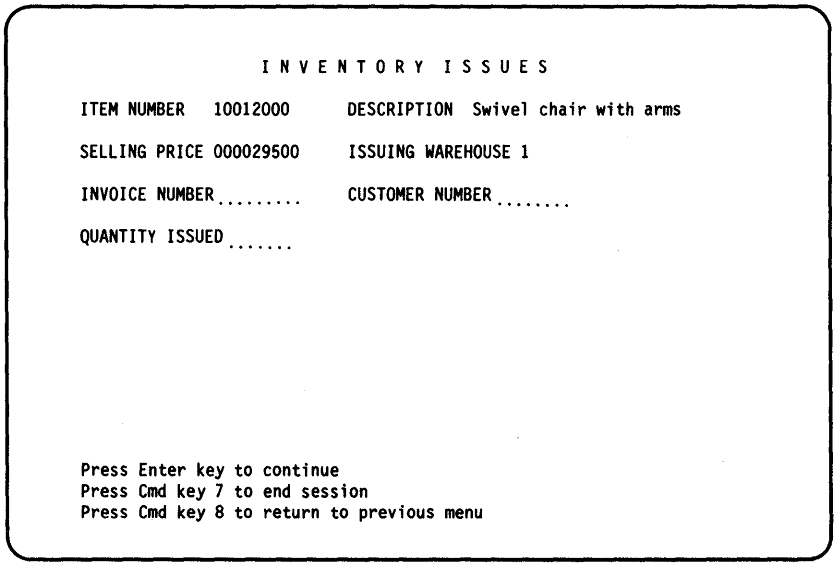
```
     INVENTORY ADJUSTMENTS/ISSUES/RECEIPTS

                         SELECT FUNCTION _

                            ADJUSTMENTS -A
                            ISSUES      -I
                            RECEIPTS    -R

                         ITEM NUMBER ........







      Press Cmd key 7 to end session
```

As you type data in a field, the cursor moves from left to right across the field.

For some fields that you fill, the cursor automatically jumps to the next input area when you have typed the last character in the field.

For fields that you do not fill, or for fields that the cursor does not automatically exit, you will have to press one of the field exit keys (Field Exit, Field+, or Field-) to leave the field. Numeric fields are right-adjusted when the field is entered; alphameric fields are left-adjusted (unless the program specifies otherwise).

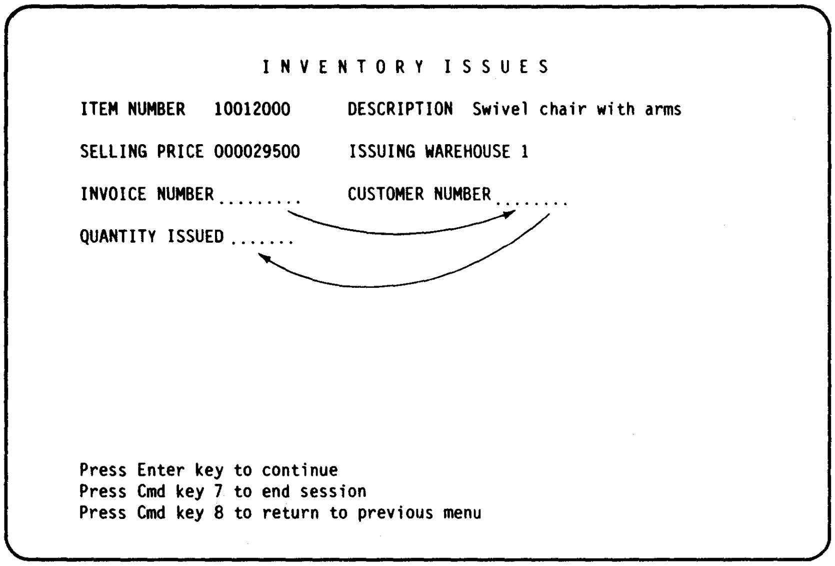
On the display below, the numeric fields are right-adjusted and filled with zeros when the Field Exit key is pressed.



```
                     INVENTORY ISSUES

      ITEM NUMBER   10012000      DESCRIPTION  Swivel chair with arms

      SELLING PRICE 000029500     ISSUING WAREHOUSE 1

      INVOICE NUMBER.........     CUSTOMER NUMBER........

      QUANTITY ISSUED.......








      Press Enter key to continue
      Press Cmd key 7 to end session
      Press Cmd key 8 to return to previous menu
```

To enter a negative value in a field, enter the value and then press the Field- key. The field will be right-adjusted and the minus sign will follow the last digit. For example, if you enter 50 and press the Field- key, 50- is displayed.

As you enter fields, the cursor advances from left to right across a line as shown below. When the rightmost input field has been entered, the cursor advances to the leftmost input field on the next lower line.



```
                  I N V E N T O R Y   I S S U E S

ITEM NUMBER    10012000       DESCRIPTION  Swivel chair with arms

SELLING PRICE 000029500       ISSUING WAREHOUSE 1

INVOICE NUMBER.........       CUSTOMER NUMBER........

QUANTITY ISSUED .......





Press Enter key to continue
Press Cmd key 7 to end session
Press Cmd key 8 to return to previous menu
```

You can use any field exit key to skip over a field, if a value in that field is not required by the program.

When you complete a display, you usually enter it by pressing the Enter key or a command key that the programmer has specified in the program. Some programs, however, enter a display automatically as soon as you type the last character in the last input field.

Reviewing Records

If the program you are working with allows you to review records in the transaction file, you can select review mode by pressing one of the following:

- Page Backward Record key (Roll ↓)
- Page Forward Record key (Roll ↑)
- Page Backward Group command key (Cmd 5)
- Page Forward Group command key (Cmd 6)
- WSU Display command key (Cmd 1) and specifying a review record number on the WSU display.

You can review records by pressing one of the following:

- Page Backward Group command key (Cmd 5)
- Page Forward Group command key (Cmd 6)
- Page Backward Record function key (Roll ↓)
- Page Forward Record function key (Roll ↑).

You can select alternative displays (if there are any) by pressing one of the following:

- Bypass Display command key (Cmd 2)
- WSU Display command key (Cmd 1) and entering a 2-character ID on the WSU display.

You can return to enter mode (at the point entry was interrupted) by pressing the:

- Resume Entry command key (Cmd 3).

Inserting Records

If the program you are working with allows you to insert records into the transaction file, you must first:

- Be in review mode (see description of *Reviewing Records*)
- Display the record that is to come before the inserted record.

You can then insert a record by pressing the:

- Insert Mode command key (Cmd 4) to get a display that you can use to insert a record.

You can select alternative displays (if there are any) by pressing one of the following:

- Bypass Display command key (Cmd 2)
- WSU Display command key (Cmd 1) and entering a 2-character ID on the WSU display.

You can return to review mode to select another place to insert a record by pressing the:

- Resume Review command key (Cmd 15).

You can return to enter mode (at the point entry was interrupted) by pressing the:

- Resume Entry command key (Cmd 3).

Deleting Records

If the program you are working with allows you to delete records from the transaction file, you must first:

- Be in review mode (see description of *Reviewing Records*)
- Display the record that is to be deleted.

You can then delete the record by pressing the:

- Delete command key (Cmd 14), and then
- The Enter key.

You can return to review mode to select another record to be deleted by pressing the:

- Resume Review command key (Cmd 15).

You can return to enter mode (at the point entry was interrupted) by pressing the:

- Resume Entry command key (Cmd 3).

Ending a Work Session

You can end a work session by pressing one of the following:

- WSU Display command key (Cmd 1) and entering EW on line 2 of the display
- A specific user command key (if the program instructs you to do so).

You can, in an emergency, use the Attention key and the Inquiry display to end a work session; however, you may well lose some of the data you entered if you use this method!

Chapter 11. WSU Example Programs

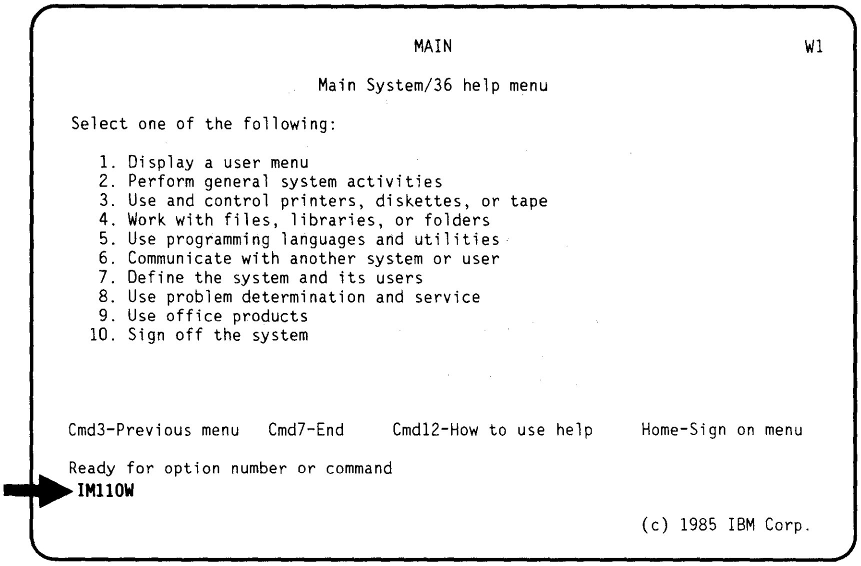
EXAMPLE PROGRAM 1: INVENTORY ADJUSTMENTS/ISSUES/RECEIPTS

This program, named IM110W, allows a maximum of four operators to concurrently enter receipts, issues, and adjustments to two master files (ITEMMSTR and ITEMBALN). Input from operators consists of selecting a function, entering an item number, and responding to requests from the display.

Output consists of a transaction file that contains all operator entered receipts, issues, and adjustments (INVTRAN).

Running IM110W

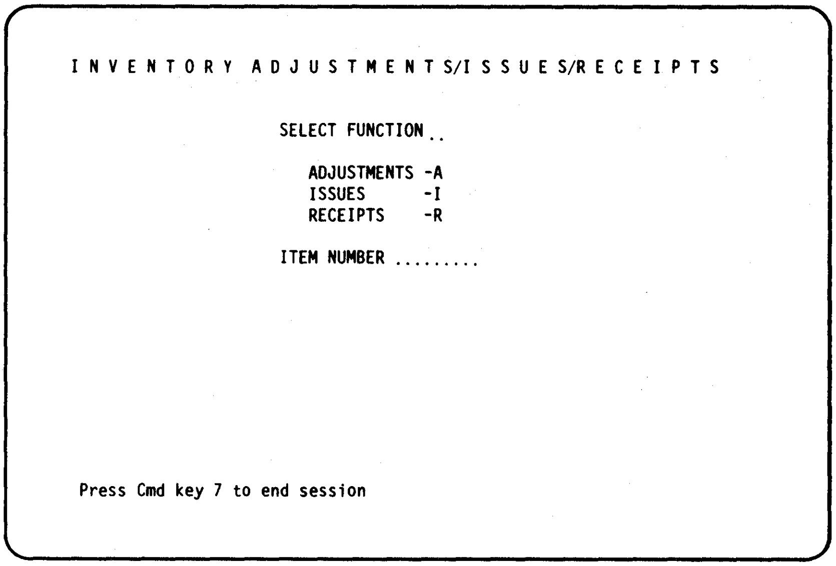
To initiate WSU program IM110W, the operator enters the procedure name, IM110W.



```
                          MAIN                                   W1

                    Main System/36 help menu

 Select one of the following:

    1. Display a user menu
    2. Perform general system activities
    3. Use and control printers, diskettes, or tape
    4. Work with files, libraries, or folders
    5. Use programming languages and utilities
    6. Communicate with another system or user
    7. Define the system and its users
    8. Use problem determination and service
    9. Use office products
   10. Sign off the system



 Cmd3-Previous menu   Cmd7-End     Cmd12-How to use help     Home-Sign on menu

 Ready for option number or command
 IM110W
                                                       (c) 1985 IBM Corp.
```

This display occurs when an operator initiates the WSU program IM110W.

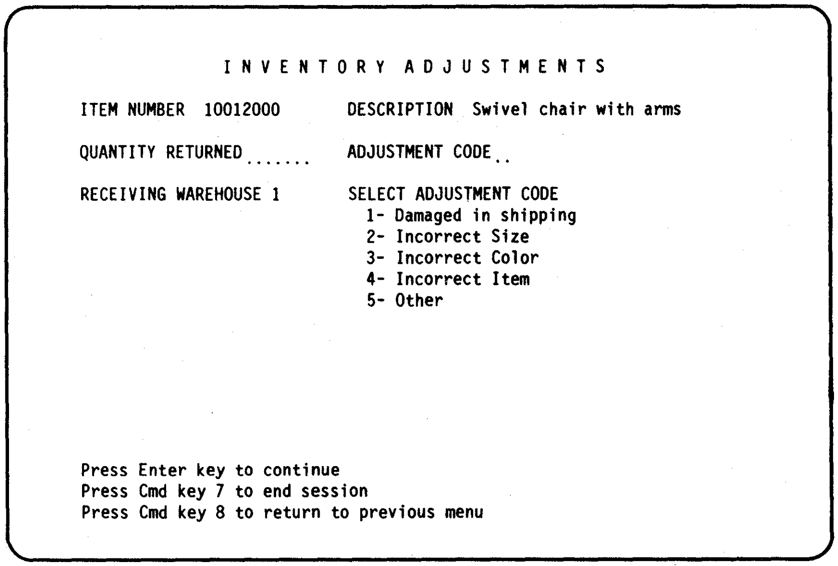
The operator enters the item number and selects one of three functions (A, I, or R). Into the selected display, the operator enters the required information, and the information is put into the transaction file INVTRAN.



```
I N V E N T O R Y   A D J U S T M E N T S/I S S U E S/R E C E I P T S

                    SELECT FUNCTION..

                       ADJUSTMENTS -A
                       ISSUES      -I
                       RECEIPTS    -R

                    ITEM NUMBER .........









 Press Cmd key 7 to end session
```

If the operator selected function A (adjustments) on the previous display, the title of the next display is Inventory Adjustments.

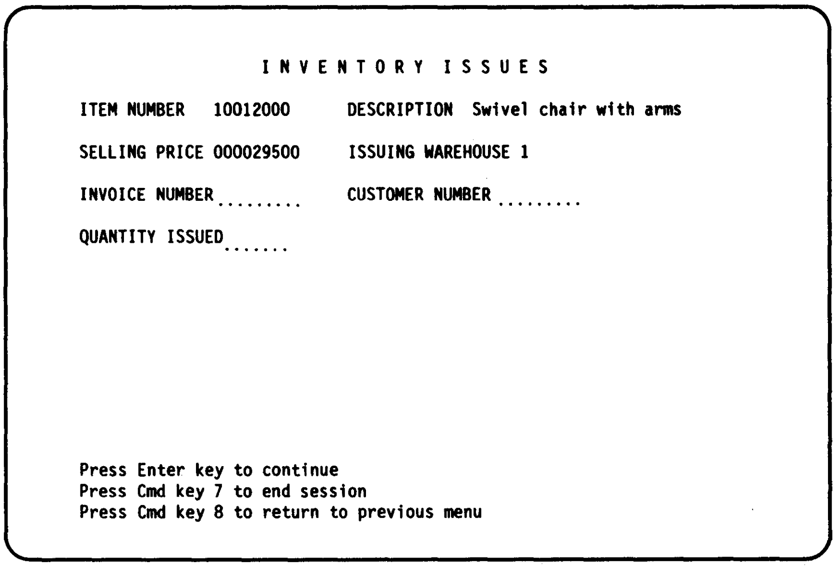
This display has two input fields: quantity returned and adjustment code. The operator enters the required data into these fields and presses the Enter key to cause processing for this display to begin. Command key 7 will end the session; command key 8 causes the previous display to be shown.



```
               I N V E N T O R Y   A D J U S T M E N T S

ITEM NUMBER  10012000       DESCRIPTION  Swivel chair with arms

QUANTITY RETURNED.......    ADJUSTMENT CODE..

RECEIVING WAREHOUSE 1       SELECT ADJUSTMENT CODE
                              1- Damaged in shipping
                              2- Incorrect Size
                              3- Incorrect Color
                              4- Incorrect Item
                              5- Other






Press Enter key to continue
Press Cmd key 7 to end session
Press Cmd key 8 to return to previous menu
```

If the operator selected function I (issues) on the previous display, the title of the next display is Inventory Issues.

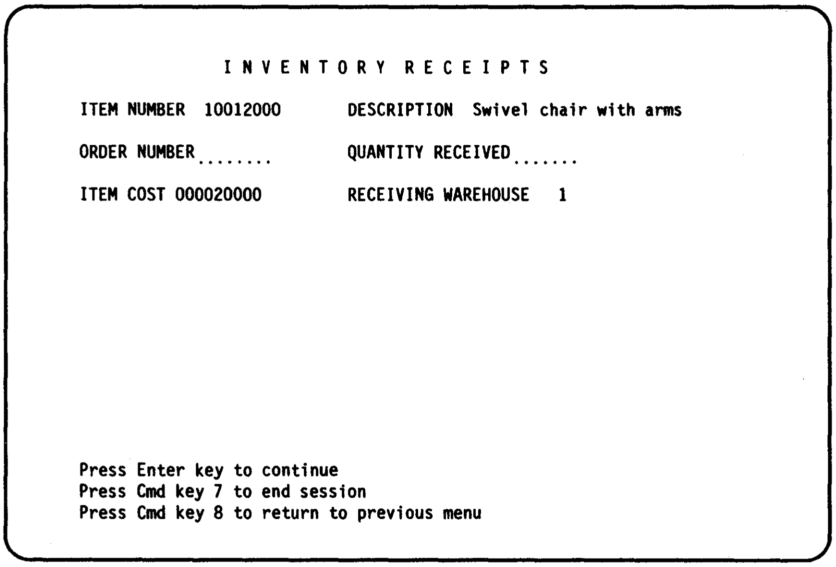
This display has three input fields: invoice number, customer number, and quantity issued. The operator enters the required data into these fields and presses the Enter key to cause processing for this display to begin. Command key 7 will end the session; command key 8 causes the previous display to be shown without any processing.



```
                  I N V E N T O R Y   I S S U E S

     ITEM NUMBER   10012000       DESCRIPTION  Swivel chair with arms

     SELLING PRICE 000029500      ISSUING WAREHOUSE 1

     INVOICE NUMBER.........      CUSTOMER NUMBER .........

     QUANTITY ISSUED......









     Press Enter key to continue
     Press Cmd key 7 to end session
     Press Cmd key 8 to return to previous menu
```

If the operator selected function R (receipts) on the previous display, the title of this display is Inventory Receipts.

This display has two input fields: order number and quantity received. The operator enters the required data into these fields and presses the Enter key to cause processing for this display to begin. Command key 7 will end the session; command key 8 causes the previous display to be shown.



```
                I N V E N T O R Y   R E C E I P T S

     ITEM NUMBER  10012000        DESCRIPTION  Swivel chair with arms

     ORDER NUMBER........         QUANTITY RECEIVED......

     ITEM COST 000020000          RECEIVING WAREHOUSE   1











     Press Enter key to continue
     Press Cmd key 7 to end session
     Press Cmd key 8 to return to previous menu
```

Coding Forms

In this example, the file definition is in three source members:

- INVTRAN contains F- and I-specifications for the transaction file, INVTRAN.
- FILESRC contains F- and I-specifications for the item master file, ITEMMSTR.
- FILESRC contains F- and I-specifications for the item balance master file, ITEMBALN.

Figure 11-1 shows the file definition for example program 1.

File Description Specifications

For the valid entries for a system, refer to the RPG reference manual for that system.

Line (3-5)	Form Type (6)	Filename (7-14)	File Type (15)	File Designation (16)	End of File (17)	Sequence (18)	File Format (19)	Block Length (20-23)	Record Length (24-27)	Mode of Processing (28)	Length of Key Field or of Record Address Field (29-30)	Record Address Type (31)	Type of File Organization or Additional Area (32)	Overflow Indicator (33-34)	Key Field Starting Location (35-38)	Extension Code E/L (39)	Device (40-46)	Symbolic Device (47-52)	Labels S/N/E/M (53)	Name of Label Exit (54-59)	Storage Index (60-65)	File Addition/Unordered (66)	Number of Tracks for Cylinder Overflow (67-68)	Number of Extents (69-70)	Tape Rewind (71)	File Condition U1-U8, UC (72-73)
02	F	INVTRAN	U						91																	
03	F	ITEMBALN	U						128		8		I		2		DISK									
04	F	ITEMMSTR	U						128		8		I		2		DISK									
05	F																									
06	F																									

IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Program | Keying Instruction | Graphic | Key | Card Electro Number | Page 1 2 of | Program Identification 75 76 77 78 79 80

Programmer | Date

Line (3-5)	Form Type (6)	Filename or Record Name (7-14)	Sequence (15-16)	Number (1/N), E (17)	Option (O), U, S (18)	Record Identifying Indicator, **, or DS (19-20)	Record Identification Codes 1 (21-27)	Record Identification Codes 2 (28-34)	Record Identification Codes 3 (35-41)	Stacker Select (42)	P/B/L/R (43)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	RPG Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields or Chaining Fields (61-62)	Field Record Relation (63-64)	Field Indicators Plus (65-66)	Field Indicators Minus (67-68)	Field Indicators Zero or Blank (69-70)	(71-74)
01	I	ITEMBALN				01																
02	I											1	1		ARCOD							
03	I											2	9		ITNBR							
04	I											10	10		WRHSE							
05	I										P	11	14	0	STOCK							
06	I										P	15	18	0	ONORD							
07	I										P	19	22	0	AVAIL							
08	I										P	23	26	0	REORD							
09	I										P	27	30	0	RENUM							
10	I										P	31	34	0	QUSMO							
11	I										P	35	38	0	QRCMO							
12	I										P	39	42	0	QADMO							
13	I										P	43	46	0	QUSYR							
14	I										P	47	50	0	QRCYR							
15	I										P	51	54	0	QADYR							
16	I										P	55	58	0	MDATE							
17	I										P	59	62	0	RRFST							
18	I										P	63	66	0	RRLST							
19	I											67	128		BUFFR							
20	I																					
	I																					
	I																					
	I																					
	I																					
	I																					

*Number of sheets per pad may vary slightly.

Figure 11-1 (Part 1 of 3). File Definition for Example Program 1

IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Program

Programmer

Date

Keying Instruction

Graphic

Key

Card Electro Number

Page of

Program Identification

Line	Form Type	Filename or Record Name	Record Identifying Indicator, **, or DS	P/B/L/R	From	To	Decimal Positions	RPG Field Name
01	I	ITEMMSTR	01					
02	I				1	1		ARCOD
03	I				2	9		ITNBR
04	I				10	39		ITDSC
05	I				40	40		ITYPE
06	I				41	42		ITCLS
07	I				43	47		WHSLC
08	I			P	48	51	0	WEGHT
09	I			P	52	56	2	UCOST
10	I			P	57	61	2	PRICE
11	I			P	62	65	0	MDATE
12	I			P	66	69	0	RRFST
13	I			P	70	73	0	RRLST
14	I				74	128		BUFFR

Figure 11-1 (Part 2 of 3). File Definition for Example Program 1

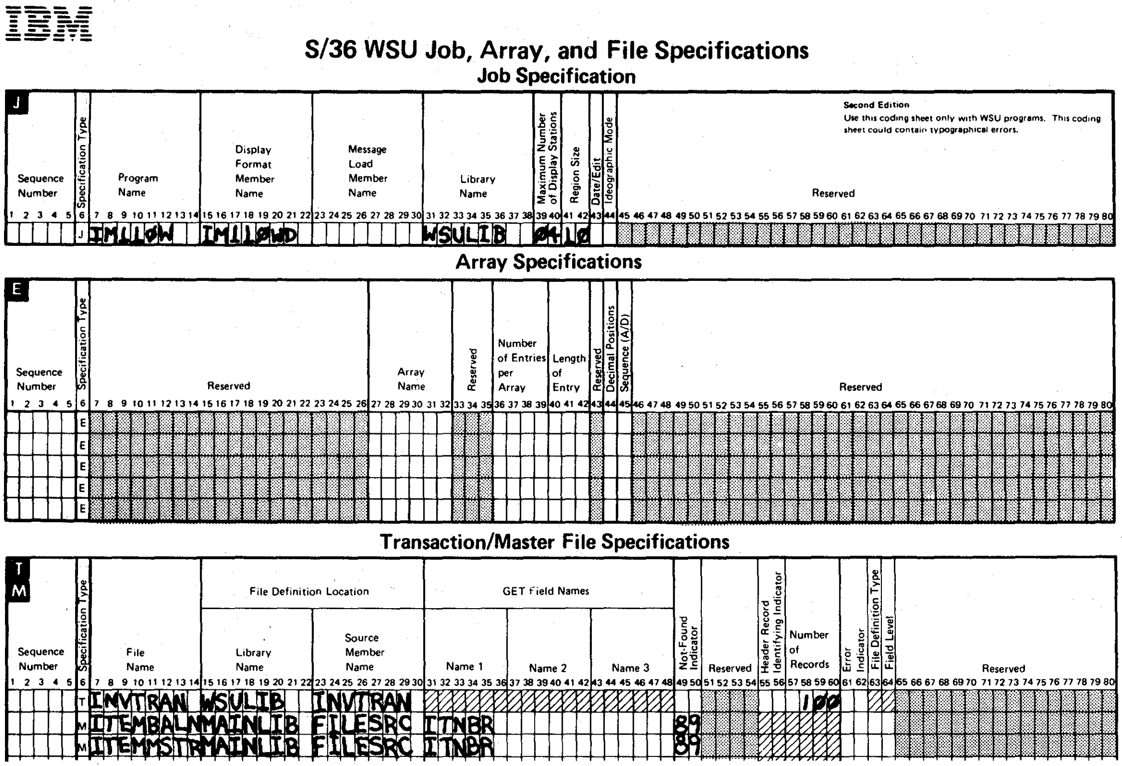
Line	Form Type	Filename or Record Name	Sequence	Number (1/N), E	Option (O), U, S	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	From	To	Decimal Positions	RPG Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank
01	I	INVTRAN				02	2		C	R												
02	I												1	1		RCODE						
03	I												2	9		ITNBR						
04	I												10	15	0	QTYOR						
05	I												16	23		CUSNO						
06	I												24	31		INVNO						
07	I												32	37		TDATE						
08	I											P	38	42	2	PRICE						
09	I											P	43	47	2	TCOST						
10	I												48	48		WRHSE						
11	I												49	78		ITDSC						
12	I					03																
13	I												1	1		RCODE						
14	I												2	3		ITCLS						
15	I												4	11		ITNBR						
16	I											P	12	16	2	PRICE						
17	I											P	17	20	0	PONBR						
18	I												21	26	0	QTYRC						
19	I												27	32		TDATE						
20	I												33	33		WRHSE						
	I												34	63		ITDSC						
	I					01																
	I												1	1		RCODE						
	I												2	9		ITNBR						
	I												10	10		ACODE						

*Number of sheets per pad may vary slightly.

Line	Form Type	Filename or Record Name	Sequence	Number (1/N), E	Option (O), U, S	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	From	To	Decimal Positions	RPG Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank
01	I												11	16	0	QTYIS						
02	I												17	22		TDATE						
03	I												23	23		WRHSE						
04	I												24	53		ITDSC						

Figure 11-1 (Part 3 of 3). File Definition for Example Program 1

The source program is in a source member named IM110W in WSULIB. Figure 11-2 shows the WSU specifications for IM110W.



IBM

S/36 WSU Job, Array, and File Specifications

Job Specification

Second Edition
Use this coding sheet only with WSU programs. This coding sheet could contain typographical errors.

Sequence Number 1-5	Specification Type 6	Program Name 7-14	Display Format Member Name 15-22	Message Load Member Name 23-30	Library Name 31-38	Maximum Number of Display Stations 39-40	Region Size 41-42	Date/Edit 43	Ideographic Mode 44	Reserved 45-80
	J	IM110W	IM110WD		WSULIB	04	10			

Array Specifications

Sequence Number 1-5	Specification Type 6	Reserved 7-26	Array Name 27-32	Reserved 33-35	Number of Entries per Array 36-39	Length of Entry 40-42	Reserved 43	Decimal Positions 44	Sequence (A/D) 45	Reserved 46-80
	E									
	E									
	E									
	E									
	E									

Transaction/Master File Specifications

Sequence Number 1-5	Specification Type 6	File Name 7-14	File Definition Location: Library Name 15-22	File Definition Location: Source Member Name 23-30	GET Field Names: Name 1 31-36	GET Field Names: Name 2 37-42	GET Field Names: Name 3 43-48	Not-Found Indicator 49-50	Reserved 51-54	Header Record Identifying Indicator 55-56	Number of Records 57-60	Error Indicator 61-62	File Definition Type 63	Field Level 64	Reserved 65-80
	T	INVTRAN	WSULIB	INVTRAN							100				
	M	ITEMBALN	MAINLIB	FILESRC	ITNBR			89							
	M	ITEMMSTR	MAINLIB	FILESRC	ITNBR			89							

Figure 11-2 (Part 1 of 9). WSU Specifications for Example Program 1

IBM

System/36 Display Format Specifications

Display Control Specification

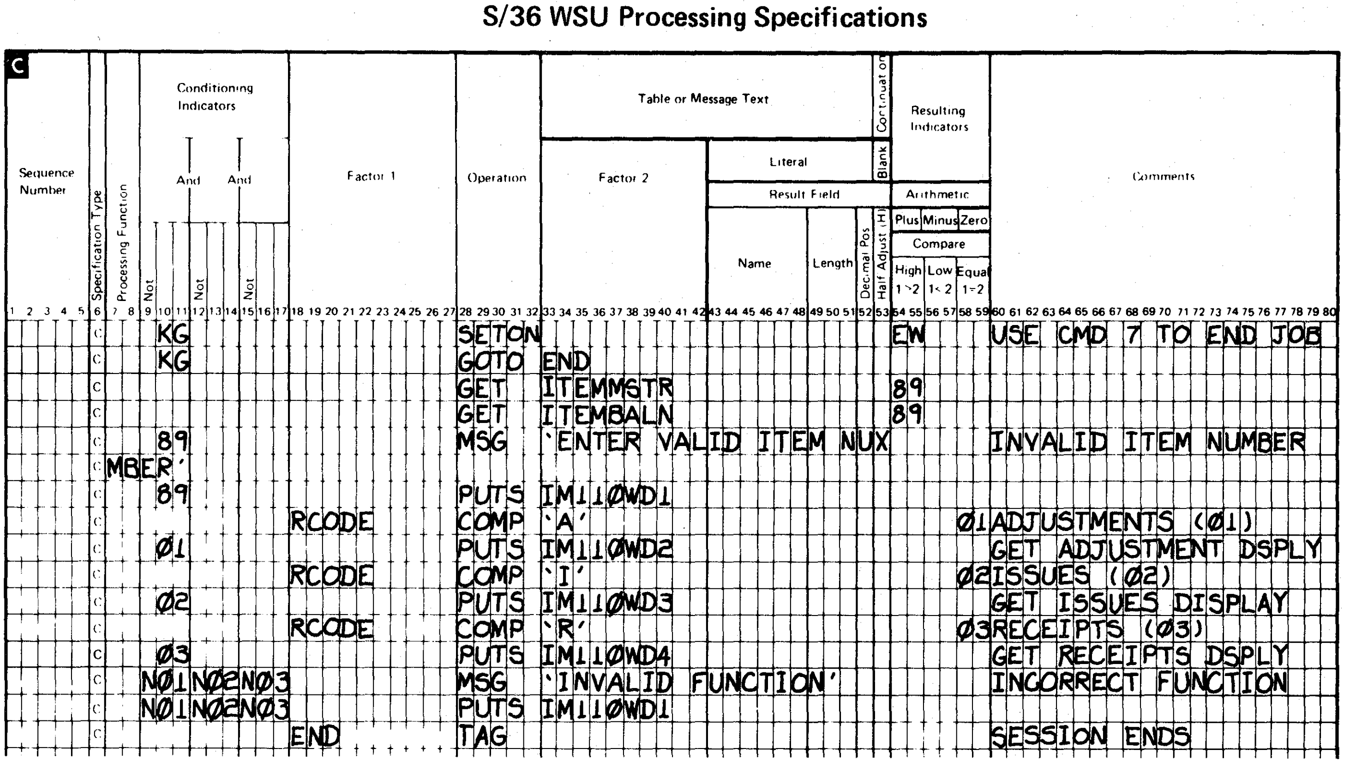
Sequence Number (1-5)	Specification Type (6)	Display Format Name (7-14)	Blink Cursor (29-30)	Enter Mode Sequence: Start (41)	Enter Mode Sequence: End (42)
	S	IM110WD1	Y	Y	Y

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

Sequence Number (1-5)	Specification Type (6)	Field Name (7-14)	Line Number (19-20)	Horizontal Position (21-22)	Input Allowed (26)	Data Type (27)	Mandatory Entry (29)	Adjust/Fill (31)	Position Cursor (32-33)	High Intensity (39-40)	Reverse Image (45-46)	Underline (47-48)	Column Separators (49)	Constant Type (56)	Constant Data (57-79)	Continuation (80)
	D		02	05						Y	Y			P	'INVENTORY AD	X
	D	JUSTMENTS/ISSUES/RECEIPTS'														
	D		05	27								Y		P	'SELECT FUNCTION'	
	D	RCODE	05	44	Y	A	Y		Y	Y			Y			
	D		07	30										P	'ADJUSTMENTS -A'	
	D		08	30										P	'ISSUES -I'	
	D		09	30										P	'RECEIPTS -R'	
	D		11	27										P	'ITEM NUMBER'	
	D	ITNBR	11	39	Y	N	Y	B					Y			
	D		22	06										P	'Press Cmd key 7 to end	X
	D	session'														
	D															
	D															
	D															
	D															

Figure 11-2 (Part 2 of 9). WSU Specifications for Example Program 1



S/36 WSU Processing Specifications

Sequence Number	Specification Type	Processing Function	Conditioning Indicators	Factor 1	Operation	Factor 2 / Table or Message Text	Continuation	Resulting Indicators: Plus / High 1>2	Minus / Low 1<2	Zero / Equal 1=2	Comments
	C		KG		SETON			EW			USE CMD 7 TO END JOB
	C		KG		GOTO	END					
	C				GET	ITEMMSTR		89			
	C				GET	ITEMBALN		89			
	C		89		MSG	'ENTER VALID ITEM NU	X				INVALID ITEM NUMBER
	C	MBER'									
	C		89		PUTS	IM110WD1					
	C			RCODE	COMP	'A'				01	ADJUSTMENTS (01)
	C		01		PUTS	IM110WD2					GET ADJUSTMENT DSPLY
	C			RCODE	COMP	'I'				02	ISSUES (02)
	C		02		PUTS	IM110WD3					GET ISSUES DISPLAY
	C			RCODE	COMP	'R'				03	RECEIPTS (03)
	C		03		PUTS	IM110WD4					GET RECEIPTS DSPLY
	C		N01N02N03		MSG	'INVALID FUNCTION'					INCORRECT FUNCTION
	C		N01N02N03		PUTS	IM110WD1					
	C			END	TAG						SESSION ENDS

Figure 11-2 (Part 3 of 9). WSU Specifications for Example Program 1

IBM

System/36 Display Format Specifications

First Edition

Use this coding sheet only to define display formats for WSU and $SFGR. This coding sheet could contain typographical errors.

Display Control Specification

S Sequence Number 1-5	Specification Type 6	Display Format Name 7-14	Format ID (WSU Only) 15-16	Start Line Number 17-18	Number of Lines to Clear 19-20	Lowercase 21	Return Input 22	Reset Keyboard (WSU Only) 23-24	Sound Alarm 25-26	Enable Function Keys 27	Enable Command Keys 28	Blink Cursor 29-30	Erase Input Fields 31-32	Override Fields 33-34	Suppress Input 35-36	Null Fill 37-38	Reserved 39	Right-to-Left Display 40	WSU Only: Enter Mode Sequence Start 41	End 42	Entry Required 43	Repeat 44	Reserved 45	Priority 46	Preprocess 47	Review Mode Record Identifying Indicators 1 48-49	2 50-51	3 52-53	Insert Mode Record Identifying Indicators 1 54-55	2 56-57	3 58-59	Reserved 60-63	Key Mask 64-79	Reserved 80
	S	IM110WD2	02			Y						Y									Y				Y									

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

D Sequence Number 1-5	Specification Type 6	WSU Field Name / Field Name 7-14	WSU Arrays: Elements per Row or Column / Field Length 15	Fill Direction 16	Line Skip Factor 17	Horizontal Positions Space Factor 18	Starting Location: Line Number 19-20	Horizontal Position 21-22	Output Data 23-24	Edit Code (WSU Only) 25	Input Allowed 26	Data Type 27	Mandatory Fill 28	Mandatory Entry 29	Self-Check 30	Adjust/Fill 31	Position Cursor 32-33	Enable Dup 34	Controlled Field Exit 35	Auto Record Advance 36	Protect Field 37-38	High Intensity 39-40	Blink Field 41-42	Nondisplay 43-44	Reverse Image 45-46	Underline 47-48	Column Separators 49	Reserved 50	Lowercase 51	Reserved 52-55	Constant Type 56	Constant Data 57-79	Continuation 80
	D						02	21														Y			Y						P	'I N V E N T O R Y A D	'
	D	J U S T M E N T S'																															
	D						04	06																							P	'ITEM NUMBER'	
	D	ITNBR					04	19	Y																								
	D						04	34																							P	'DESCRIPTION'	
	D	ITDSC					04	47	Y																								
	D						06	06																							P	'QUANTITY RETURNED'	
	D	QTYRC					06	24			Y			Y		Z	Y										Y						
	D						06	34																							P	'ADJUSTMENT CODE'	
	D	ACODE					06	50			Y	N		Y													Y						
	D						08	06																							P	'RECEIVING WAREHOUSE'	
	D	WRHSE					08	26	Y																								
	D						08	34																							P	'SELECT ADJUSTMENT CODE	X
	D	'																															
	D						09	36																							P	'1- Damaged in shipping	X
	D	'																															
	D						10	36																							P	'2- Incorrect size'	
	D						11	36																							P	'3- Incorrect color'	
	D						12	36																							P	'4- Incorrect item'	
	D						13	36																							P	'5- Other'	
	D						21	06																							P	'Press Enter key to con	X
	D	tinue'																															
	D						22	06																							P	'Press Cmd key 7 to end	X
	D	session'																															
	D						23	06																							P	'Press Cmd key 8 to ret	X
	D	urn to previous menu'																															
	D																																
	D																																
	D																																
	D																																

Figure 11-2 (Part 4 of 9). WSU Specifications for Example Program 1

S/36 WSU Processing Specifications

Sequence Number 1–5	Specification Type 6	Processing Function 7–8	Conditioning Indicators 9–11 (Not)	And 12–14 (Not)	And 15–17 (Not)	Factor 1 18–27	Operation 28–32	Factor 2 / Table or Message Text 33–42	Literal / Result Field Name 43–48	Length 49–51	Decimal Pos 52	Half Adjust (H) 53	Resulting Indicators Plus / High 1>2 54–55	Minus / Low 1<2 56–57	Zero / Equal 1=2 58–59	Comments 60–80
	C		KG				SETON						EW			CMD 7 TO END SESSION
	C		KG				GOTO	END1								
	C		KH				PUTS	IM110WD1								
	C						GET	ITEMMSTR								
	C						GET	ITEMBALN								
	C						PUTS	IM110WD2								
	C					ACODE	COMP	'1';'2';'3';'4';'5'							72	
	C		N72				MSG	'INVALID CODE'								
	C		N72				PUTS	IM110WD2								
	C						MOVE	UDATE	TDATE							
	C						PUT	INVTRAN					01			
	C					END1	TAG									

Figure 11-2 (Part 5 of 9). WSU Specifications for Example Program 1

System/36 Display Format Specifications

First Edition

Use this coding sheet only to define display formats for WSU and $SFGR. This coding sheet could contain typographical errors.

Display Control Specification

Sequence Number (1-5)	Specification Type (6)	Display Format Name (7-14)	Format ID (WSU Only) (15-16)	Start Line Number (17-18)	Number of Lines to Clear (19-20)	Lowercase (21)	Return Input (22)	Reset Keyboard (WSU Only) (23-24)	Sound Alarm (25-26)	Enable Function Keys (27)	Enable Command Keys (28)	Blink Cursor (29-30)	Erase Input Fields (31-32)	Override Fields (33-34)	Suppress Input (35-36)	Null Fill (37-38)	Reserved (39)	Right-to-Left Display (40)	WSU Only: Enter Mode Sequence Start (41)	End (42)	Entry Required (43)	Repeat (44)	Reserved (45)	Priority (46)	Preprocess (47)	Review Mode Record Identifying Indicators 1 (48-50)	2 (51-53)	3	Insert Mode Record Identifying Indicators 1 (54-56)	2 (57-59)	3	Reserved (60-63)	Key Mask (64-79)	Reserved (80)
	S	IM110WD3										Y									Y				Y									

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

Sequence Number (1-5)	Specification Type (6)	WSU Field Name / Field Name (7-14)	WSU Arrays (Elements per Row or Column, Fill Direction, Line Skip Factor, Horizontal Positions Space Factor) / Field Length (15-18)	Starting Location: Line Number (19-20)	Horizontal Position (21-22)	Output Data (23-24)	Edit Code (WSU Only) (25)	Input Allowed (26)	Data Type (27)	Mandatory Fill (28)	Mandatory Entry (29)	Self-Check (30)	Adjust/Fill (31)	Position Cursor (32-33)	Enable Dup (34)	Controlled Field Exit (35)	Auto Record Advance (36)	Protect Field (37-38)	High Intensity (39-40)	Blink Field (41-42)	Nondisplay (43-44)	Reverse Image (45-46)	Underline (47-48)	Column Separators (49)	Reserved (50)	Lowercase (51)	Reserved (52-55)	Constant Type (56)	Constant Data (57-79)	Continuation (80)
	D			02	25														Y			Y						P	`I N V E N T O R Y I S	X
	D	S U E S'																												
	D			04	06																							P	`ITEM NUMBER'	
	D	ITNBR		04	20	Y																								
	D			04	34																							P	`DESCRIPTION'	
	D	ITDSC		04	47	Y																								
	D			06	06																							P	`SELLING PRICE'	
	D	PRICE		06	20	Y																								
	D			06	34																							P	`ISSUING WAREHOUSE'	
	D	WRHSE		06	52	Y																								
	D			08	06																							P	`INVOICE NUMBER'	
	D	INVNO		08	21			Y			Y		Z	Y										Y						
	D			08	34																							P	`CUSTOMER NUMBER'	
	D	CUSNO		08	50			Y			Y		Z											Y						
	D			10	06																							P	`QUANTITY ISSUED'	
	D	QTYIS		10	23			Y			Y		Z											Y						
	D			21	06																							P	`Press Enter key to con	X
	D	tinue'																												
	D			22	06																							P	`Press Cmd key 7 to end	X
	D	session'																												
	D			23	06																							P	`Press Cmd key 8 to ret	X
	D	urn to previous menu'																												

Figure 11-2 (Part 6 of 9). WSU Specifications for Example Program 1

S/36 WSU Processing Specifications

Sequence Number (1-5)	Specification Type (6)	Processing Function (7-8)	Conditioning Indicators: Not (9)	Conditioning Indicators (10-11)	And: Not (12)	And (13-14)	And: Not (15)	And (16-17)	Factor 1 (18-27)	Operation (28-32)	Table or Message Text / Factor 2 (33-42)	Literal / Result Field: Name (43-48)	Length (49-51)	Decimal Pos (52)	Half Adjust (H) (53)	Resulting Indicators: Plus / High 1>2 (54-55)	Minus / Low 1<2 (56-57)	Zero / Equal 1=2 (58-59)	Comments (60-80)
	C			KG						SETON						EW			CMD 7 TO END SESSION
	C			KG						GOTO	END2								
	C			KH						PUTS	IM110WD1								
	C									GET	ITEMMSTR								MASTER FILE (ITEM)
	C									GET	ITEMBALN								MASTER FILE (BALANCE)
	C									PUTS	IM110WD3								
	C								QTYIS	MULT	PRICE	TCOST							
	C									MOVE	UDATE	TDATE							DATE TO TRANS FILE
	C									PUT	INVTRAN					02			WRITE TO TRANS FILE
	C								END2	TAG									END SESSION

Figure 11-2 (Part 7 of 9). WSU Specifications for Example Program 1

System/36 Display Format Specifications

Display Control Specification

Sequence Number 1-5	Specification Type 6	Display Format Name 7-14	Format ID (WSU Only) 15-16	Start Line Number 17-18	Number of Lines to Clear 19-20	Lowercase 21	Return Input 22	Reset Keyboard (WSU Only) 23-24	Sound Alarm 25-26	Enable Function Keys 27	Enable Command Keys 28	Blink Cursor 29-30	Erase Input Fields 31-32	Override Fields 33-34	Suppress Input 35-36	Null Fill 37-38	Reserved 39	Right-to-Left Display 40	WSU Only: Enter Mode Sequence Start 41	WSU Only: Enter Mode Sequence End 42	WSU Only: Entry Required 43	WSU Only: Repeat 44	WSU Only: Reserved 45	WSU Only: Priority 46	WSU Only: Preprocess 47	WSU Only: Review Mode Record Identifying Indicators 1 48-49	2 50-51	3 52-53	WSU Only: Insert Mode Record Identifying Indicators 1 54-55	2 56-57	3 58-59	Reserved 60-63	Key Mask 64-79	Reserved 80
	S	IM110WD4										Y									Y				Y									

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

Sequence Number 1-5	Specification Type 6	WSU Field Name / Field Name 7-14	WSU Arrays (Elements per Row or Column, Fill Direction, Line Skip Factor, Horizontal Positions Space Factor) / Field Length 15-18	Starting Location: Line Number 19-20	Starting Location: Horizontal Position 21-22	Output Data 23-24	Edit Code (WSU Only) 25	Input Allowed 26	Data Type 27	Mandatory Fill 28	Mandatory Entry 29	Self-Check 30	Adjust/Fill 31	Position Cursor 32-33	Enable Dup 34	Controlled Field Exit 35	Auto Record Advance 36	Protect Field 37-38	High Intensity 39-40	Blink Field 41-42	Nondisplay 43-44	Reverse Image 45-46	Underline 47-48	Column Separators 49	Reserved 50	Lowercase 51	Reserved 52-55	Constant Type 56	Constant Data 57-79	Continuation 80
	D			02	21														Y			Y						P	'I N V E N T O R Y R E	X
	D	C E I P T S'																												
	D			04	06																							P	'ITEM NUMBER'	
	D	ITNBR		04	19	Y																								
	D			04	34																							P	'DESCRIPTION'	
	D	ITDSC		04	47	Y																								
	D			06	06																							P	'ORDER NUMBER'	
	D	PONBR		06	19			Y			Y		Z	Y										Y						
	D			06	34																							P	'QUANTITY RECEIVED'	
	D	QTYRC		06	53			Y			Y		Z											Y						
	D			08	06																							P	'ITEM COST'	
	D	UCOST		08	16	Y																								
	D			08	34																							P	'RECEIVING WAREHOUSE'	
	D	WRHSE		08	56	Y																								
	D			21	06																							P	'Press Enter key to con	X
	D	tinue'																												
	D			22	06																							P	'Press Cmd key 7 to end	X
	D	session'																												
	D			23	06																							P	'Press Cmd key 8 to ret	X
	D	urn to previous menu'																												

Figure 11-2 (Part 8 of 9). WSU Specifications for Example Program 1

S/36 WSU Processing Specifications

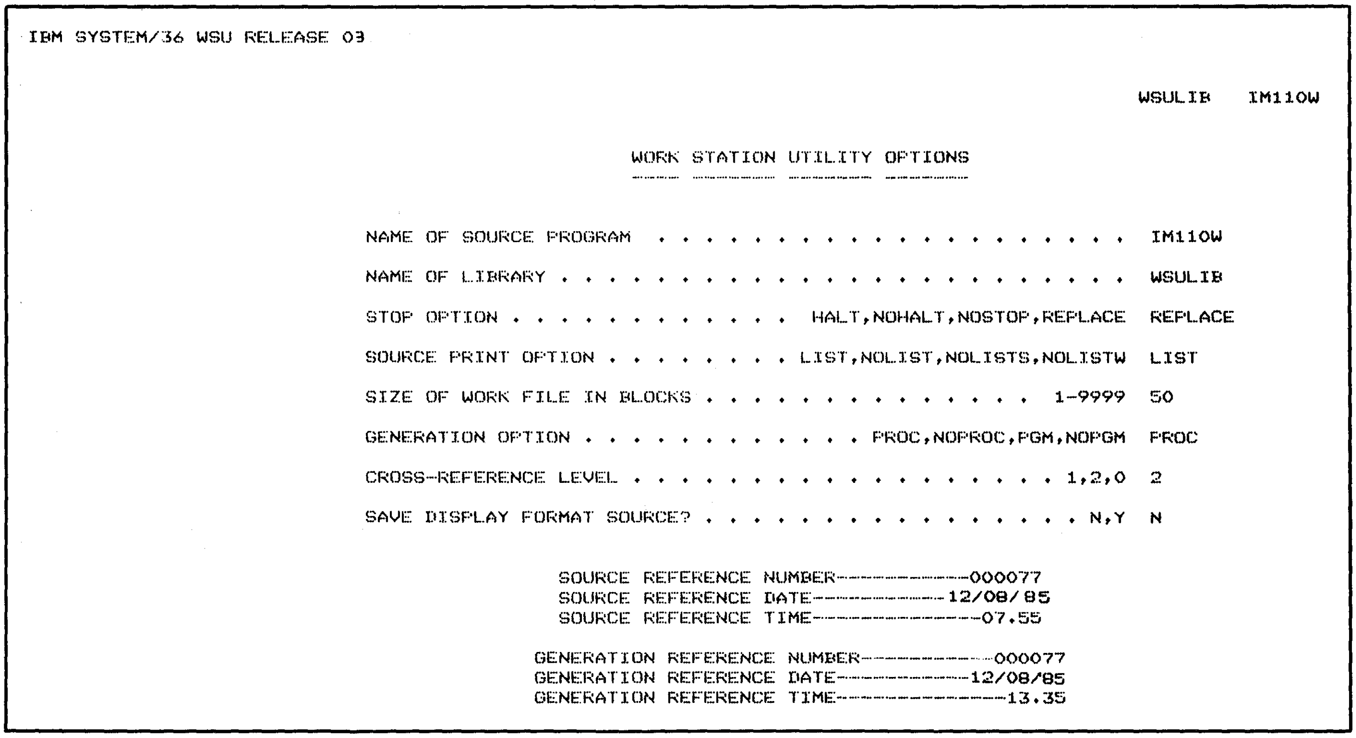
C

Sequence Number (1-5)	Specification Type (6)	Processing Function (7-8)	Conditioning Indicators: Not (9)	(10-11)	And: Not (12)	(13-14)	And: Not (15)	(16-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Literal / Result Field: Name (43-48)	Length (49-51)	Decimal Pos (52)	Half Adjust (H) (53)	Continuation / Blank	Resulting Indicators: Plus / High 1>2 (54-55)	Minus / Low 1<2 (56-57)	Zero / Equal 1=2 (58-59)	Comments (60-80)
	C			KG						SETON							EW			TO END SESSION
	C			KG						GOTO	END3									
	C			KH						PUTS	IM110WD1									
	C									GET	ITEMMSTR									MASTER FILE (ITEM)
	C									GET	ITEMBALN									MASTER FILE (BALANCE)
	C									PUTS	IM110WD4									
	C									MOVE	UDATE	TDATE								DATE TO TRANS FILE
	C									PUT	INVTRAN						03			WRITE TO TRANS FILE
	C								END3	TAG										END SESSION

Figure 11-2 (Part 9 of 9). WSU Specifications for Example Program 1

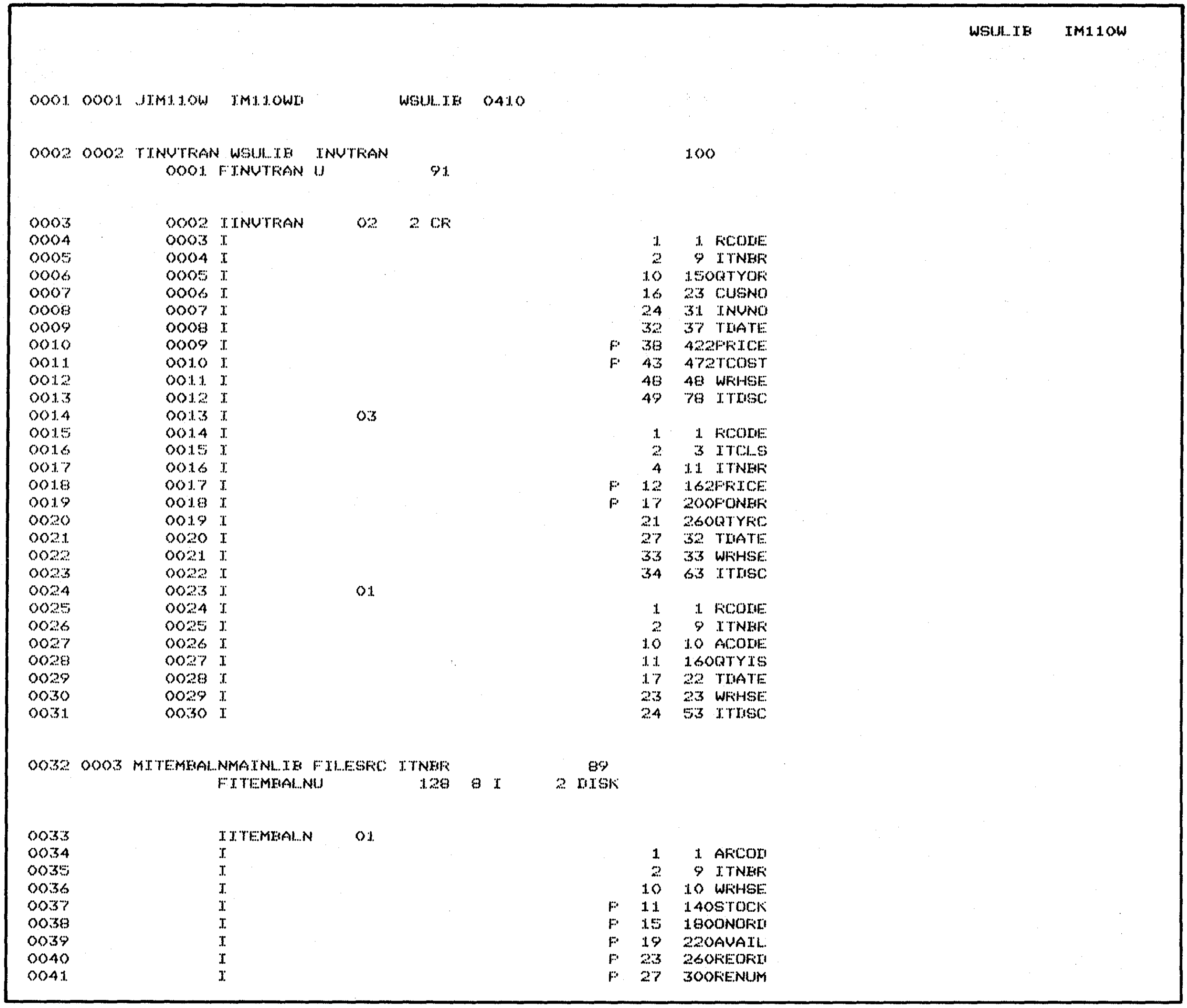
Program Generation Printout

Figure 11-3 shows the generation printout of IM110W.



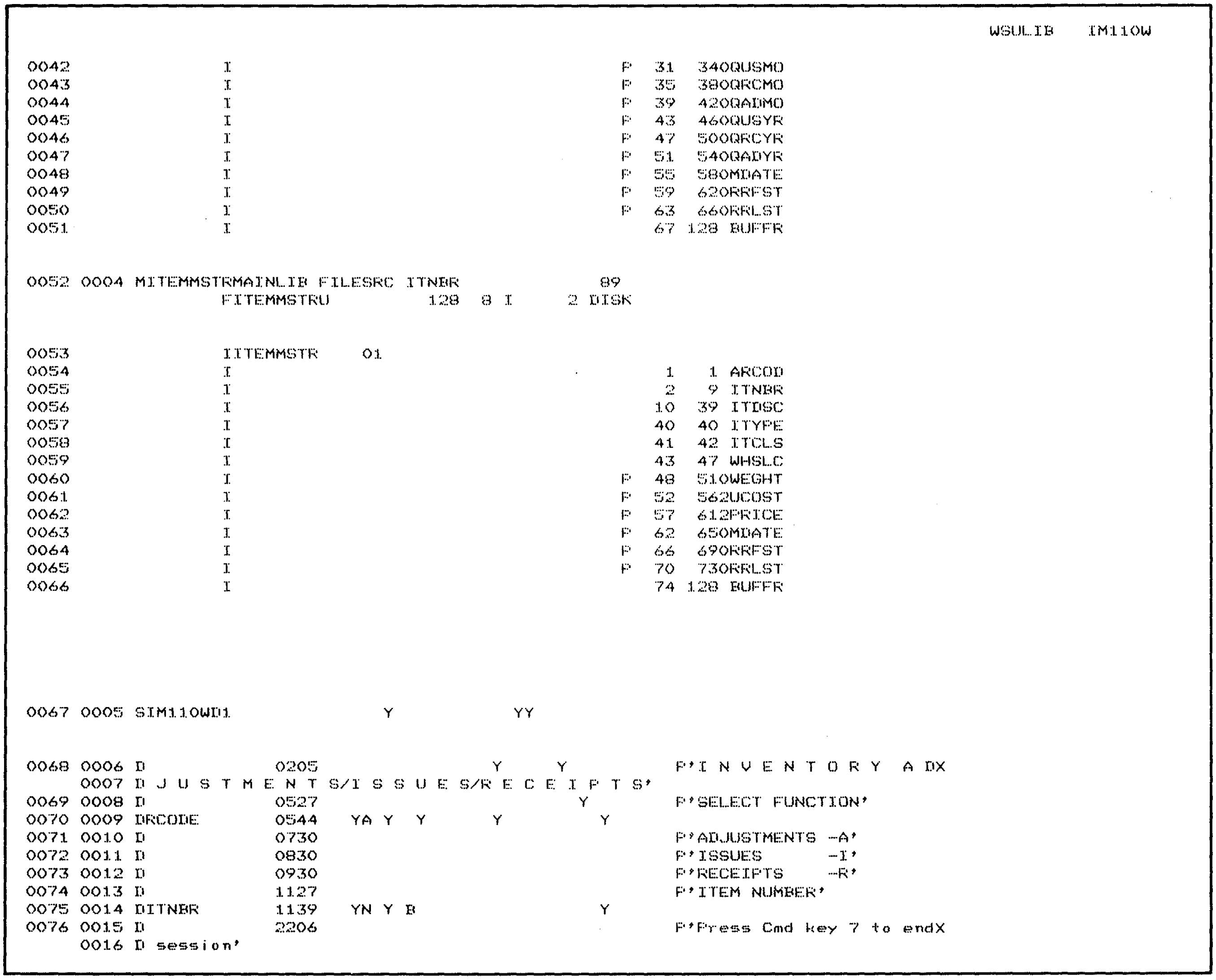
```
IBM SYSTEM/36 WSU RELEASE 03

                                                                                                         WSULIB    IM110W

                                                                   WORK STATION UTILITY OPTIONS
                                                                   ---- ------- ------- -------

                                          NAME OF SOURCE PROGRAM  . . . . . . . . . . . . . . . . . . . . .  IM110W

                                          NAME OF LIBRARY . . . . . . . . . . . . . . . . . . . . . . . . .  WSULIB

                                          STOP OPTION . . . . . . . . . . . .  HALT,NOHALT,NOSTOP,REPLACE  REPLACE

                                          SOURCE PRINT OPTION . . . . . . . .  LIST,NOLIST,NOLISTS,NOLISTW  LIST

                                          SIZE OF WORK FILE IN BLOCKS . . . . . . . . . . . . . .  1-9999  50

                                          GENERATION OPTION . . . . . . . . . . . .  PROC,NOPROC,PGM,NOPGM  PROC

                                          CROSS-REFERENCE LEVEL . . . . . . . . . . . . . . . . . .  1,2,0  2

                                          SAVE DISPLAY FORMAT SOURCE? . . . . . . . . . . . . . . . .  N,Y  N

                                                          SOURCE REFERENCE NUMBER-------------000077
                                                          SOURCE REFERENCE DATE--------------12/08/85
                                                          SOURCE REFERENCE TIME----------------07.55

                                                        GENERATION REFERENCE NUMBER-------------000077
                                                        GENERATION REFERENCE DATE--------------12/08/85
                                                        GENERATION REFERENCE TIME----------------13.35
```

Figure 11-3 (Part 1 of 13). Generation Printout for Example Program 1



```
                                                                                          WSULIB   IM110W


0001 0001 JIM110W  IM110WD         WSULIB  0410


0002 0002 TINVTRAN WSULIB  INVTRAN                            100
             0001 FINVTRAN U          91


0003         0002 IINVTRAN     02   2 CR
0004         0003 I                                        1   1 RCODE
0005         0004 I                                        2   9 ITNBR
0006         0005 I                                       10  150QTYOR
0007         0006 I                                       16  23 CUSNO
0008         0007 I                                       24  31 INVNO
0009         0008 I                                       32  37 TDATE
0010         0009 I                                    P  38  422PRICE
0011         0010 I                                    P  43  472TCOST
0012         0011 I                                       48  48 WRHSE
0013         0012 I                                       49  78 ITDSC
0014         0013 I            03
0015         0014 I                                        1   1 RCODE
0016         0015 I                                        2   3 ITCLS
0017         0016 I                                        4  11 ITNBR
0018         0017 I                                    P  12  162PRICE
0019         0018 I                                    P  17  200PONBR
0020         0019 I                                       21  260QTYRC
0021         0020 I                                       27  32 TDATE
0022         0021 I                                       33  33 WRHSE
0023         0022 I                                       34  63 ITDSC
0024         0023 I            01
0025         0024 I                                        1   1 RCODE
0026         0025 I                                        2   9 ITNBR
0027         0026 I                                       10  10 ACODE
0028         0027 I                                       11  160QTYIS
0029         0028 I                                       17  22 TDATE
0030         0029 I                                       23  23 WRHSE
0031         0030 I                                       24  53 ITDSC


0032 0003 MITEMBALNMAINLIB FILESRC ITNBR             89
                  FITEMBALNU        128   8 I     2 DISK


0033              IITEMBALN    01
0034              I                                        1   1 ARCOD
0035              I                                        2   9 ITNBR
0036              I                                       10  10 WRHSE
0037              I                                    P  11  140STOCK
0038              I                                    P  15  180ONORD
0039              I                                    P  19  220AVAIL
0040              I                                    P  23  260REORD
0041              I                                    P  27  300RENUM
```

Figure 11-3 (Part 2 of 13). Generation Printout for Example Program 1



```
                                                                                        WSULIB   IM110W

0042              I                                    P 31  340QUSMO
0043              I                                    P 35  380QRCMO
0044              I                                    P 39  420QADMO
0045              I                                    P 43  460QUSYR
0046              I                                    P 47  500QRCYR
0047              I                                    P 51  540QADYR
0048              I                                    P 55  580MDATE
0049              I                                    P 59  620RRFST
0050              I                                    P 63  660RRLST
0051              I                                      67 128 BUFFR


0052 0004 MITEMMSTRMAINLIB FILESRC ITNBR              89
                  FITEMMSTRU         128  8 I      2 DISK


0053              IITEMMSTR    01
0054              I                                       1   1 ARCOD
0055              I                                       2   9 ITNBR
0056              I                                      10  39 ITDSC
0057              I                                      40  40 ITYPE
0058              I                                      41  42 ITCLS
0059              I                                      43  47 WHSLC
0060              I                                    P 48  510WEGHT
0061              I                                    P 52  562UCOST
0062              I                                    P 57  612PRICE
0063              I                                    P 62  650MDATE
0064              I                                    P 66  690RRFST
0065              I                                    P 70  730RRLST
0066              I                                      74 128 BUFFR




0067 0005 SIM110WD1               Y          YY


0068 0006 D            0205                Y     Y         F'I N V E N T O R Y  A DX
     0007 D J U S T M E N T S/I S S U E S/R E C E I P T S'
0069 0008 D            0527                        Y       F'SELECT FUNCTION'
0070 0009 DRCODE       0544   YA Y  Y      Y         Y     
0071 0010 D            0730                                F'ADJUSTMENTS  -A'
0072 0011 D            0830                                F'ISSUES       -I'
0073 0012 D            0930                                F'RECEIPTS     -R'
0074 0013 D            1127                                F'ITEM NUMBER'
0075 0014 DITNBR       1139   YN Y B                 Y
0076 0015 D            2206                                F'Press Cmd key 7 to endX
     0016 D session'
```

Figure 11-3 (Part 3 of 13). Generation Printout for Example Program 1

```
                                                                                         WSULIB    IM110W

0077 0017 C   KG                SETON                     EW    USE CMD 7 TO END JOB
0078 0018 C   KG                GOTO END
0079 0019 C                     GET  ITEMMSTR             89
0080 0020 C                     GET  ITEMBALN             89
0081 0021 C   89                MSG  'ENTER VALID ITEM NUX      INVALID ITEM NUMBER
     0022 CMBER'
0082 0023 C   89                PUTS IM110WD1
0083 0024 C           RCODE     COMP 'A'                      01ADJUSTMENTS (01)
0084 0025 C   01                PUTS IM110WD2                   GET ADJUSTMENT DSPLY
0085 0026 C           RCODE     COMP 'I'                      02ISSUES  (02)
0086 0027 C   02                PUTS IM110WD3                   GET ISSUES DISPLAY
0087 0028 C           RCODE     COMP 'R'                      03RECEIPTS    (03)
0088 0029 C   03                PUTS IM110WD4                   GET RECEIPTS DSPLY
0089 0030 C  N01N02N03          MSG  'INVALID FUNCTION'         INCORRECT FUNCTION
0090 0031 C  N01N02N03          PUTS IM110WD1
0091 0032 C           END       TAG                             SESSION ENDS





0092 0033 SIM110WD202    Y       Y             Y   Y


0093 0034 D            0221                Y     Y          P'I N V E N T O R Y  A DX
     0035 D J U S T M E N T S'
0094 0036 D            0406                                 P'ITEM NUMBER'
0095 0037 DITNBR       0419Y
0096 0038 D            0434                                 P'DESCRIPTION'
0097 0039 DITDSC       0447Y
0098 0040 D            0606                                 P'QUANTITY RETURNED'
0099 0041 DQTYRC       0624   Y  Y ZY                Y
0100 0042 D            0634                                 P'ADJUSTMENT CODE'
0101 0043 DACODE       0650   YN Y                   Y
0102 0044 D            0806                                 P'RECEIVING WAREHOUSE'
0103 0045 DWRHSE       0826Y
0104 0046 D            0834                                 P'SELECT ADJUSTMENT CODEX
     0047 D'
0105 0048 D            0936                                 P'1- Damaged in shippingX
     0049 D'
0106 0050 D            1036                                 P'2- Incorrect Size'
0107 0051 D            1136                                 P'3- Incorrect Color'
0108 0052 D            1236                                 P'4- Incorrect Item'
0109 0053 D            1336                                 P'5- Other'
0110 0054 D            2106                                 P'Press Enter key to conX
     0055 Dtinue'
0111 0056 D            2206                                 P'Press Cmd key 7 to endX
     0057 D session'
0112 0058 D            2306                                 P'Press Cmd key 8 to ret'
     0059 Durn to previous menu'
```

Figure 11-3 (Part 4 of 13). Generation Printout for Example Program 1

```
                                                                                         WSULIB   IM110W

0113 0060 C   KG                SETON                     EW    CMD 7 TO END SESSION
0114 0061 C   KG                GOTO END1
0115 0062 C   KH                PUTS IM110WD1
0116 0063 C                     GET  ITEMMSTR
0117 0064 C                     GET  ITEMBALN
0118 0065 C                     PUTS IM110WD2
0119 0066 C           ACODE     COMP '1';'2';'3';'4';'5'      72
0120 0067 C  N72                MSG  'INVALID CODE'
0121 0068 C  N72                PUTS IM110WD2
0122 0069 C                     MOVE UDATE     TDATE
0123 0070 C                     PUT  INVTRAN              01
0124 0071 C           END1      TAG






0125 0072 SIM110WD3              Y             Y   Y


0126 0073 D            0225                Y     Y          P'I N V E N T O R Y  I SX
     0074 D S U E S'
0127 0075 D            0406                                 P'ITEM NUMBER'
0128 0076 DITNBR       0420Y
0129 0077 D            0434                                 P'DESCRIPTION'
0130 0078 DITDSC       0447Y
0131 0079 D            0606                                 P'SELLING PRICE'
0132 0080 DPRICE       0620Y
0133 0081 D            0634                                 P'ISSUING WAREHOUSE'
0134 0082 DWRHSE       0652Y
0135 0083 D            0806                                 P'INVOICE NUMBER'
0136 0084 DINVNO       0821   Y  Y ZY                 Y
0137 0085 D            0834                                 P'CUSTOMER NUMBER'
0138 0086 DCUSNO       0850   Y  Y Z                  Y
0139 0087 D            1006                                 P'QUANTITY ISSUED'
0140 0088 DQTYIS       1023   Y  Y Z                  Y
0141 0089 D            2106                                 P'Press Enter key to conX
     0090 Dtinue'
0142 0091 D            2206                                 P'Press Cmd key 7 to endX
     0092 D session'
0143 0093 D            2306                                 P'Press Cmd key 8 to retX
     0094 Durn to previous menu'


0144 0095 C   KG                SETON                     EW    CMD 7 TO END SESSION
0145 0096 C   KG                GOTO END2
0146 0097 C   KH                PUTS IM110WD1
0147 0098 C                     GET  ITEMMSTR                   MASTER FILE  (ITEM)
0148 0099 C                     GET  ITEMBALN                   MASTER FILE (BALANCE)
0149 0100 C                     PUTS IM110WD3
0150 0101 C           QTYIS     MULT PRICE     TCOST
0151 0102 C                     MOVE UDATE     TDATE            DATE TO TRANS FILE
```

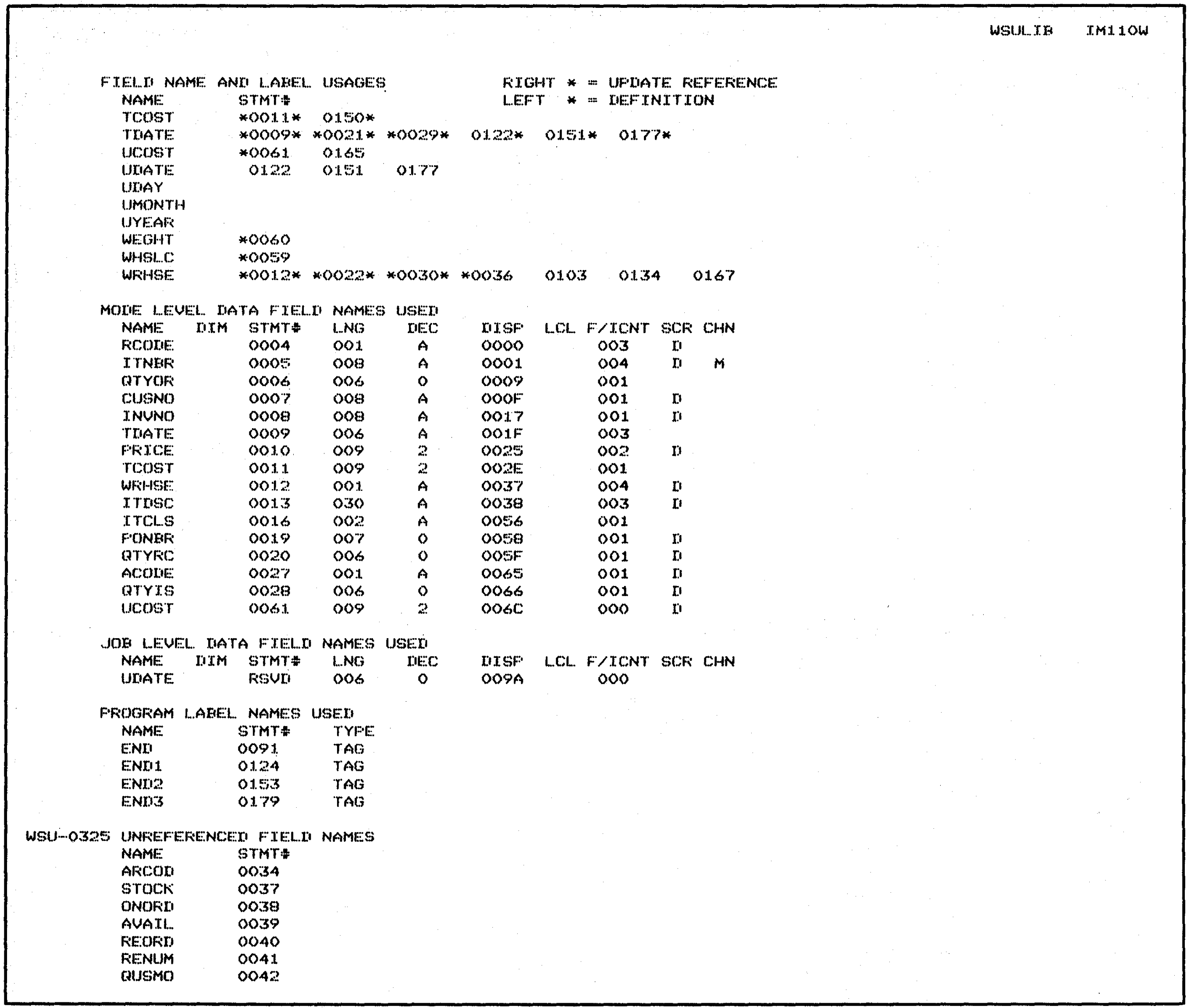
Figure 11-3 (Part 5 of 13). Generation Printout for Example Program 1

```
                                                                                        WSULIB   IM110W

0152 0103 C                     PUT  INVTRAN             02    WRITE TO TRANS FILE
0153 0104 C           END2      TAG                            END SESSION






0154 0105 SIM110WD4              Y             Y   Y


0155 0106 D            0221                Y     Y         P'I N V E N T O R Y  R EX
     0107 D C E I P T S'
0156 0108 D            0406                                P'ITEM NUMBER'
0157 0109 DITNBR       0419Y
0158 0110 D            0434                                P'DESCRIPTION'
0159 0111 DITDSC       0447Y
0160 0112 D            0606                                P'ORDER NUMBER'
0161 0113 DPONBR       0619   Y  Y ZY                Y
0162 0114 D            0634                                P'QUANTITY RECEIVED'
0163 0115 DQTYRC       0653   Y  Y Z                 Y
0164 0116 D            0806                                P'ITEM COST'
0165 0117 DUCOST       0816Y
0166 0118 D            0834                                P'RECEIVING WAREHOUSE'
0167 0119 DWRHSE       0856Y
0168 0120 D            2106                                P'Press Enter key to conX
     0121 Dtinue'
0169 0122 D            2206                                P'Press Cmd key 7 to endX
     0123 D session'
0170 0124 D            2306                                P'Press Cmd key 8 to retX
     0125 Durn to previous menu'


0171 0126 C   KG                SETON                    EW    TO END SESSION
0172 0127 C   KG                GOTO END3
0173 0128 C   KH                PUTS IM110WD1
0174 0129 C                     GET  ITEMMSTR                  MASTER FILE (ITEM)
0175 0130 C                     GET  ITEMBALN                  BALANCE FILE (MASTER)
0176 0131 C                     PUTS IM110WD4
0177 0132 C                     MOVE UDATE     TDATE           DATE TO TRANS FILE
0178 0133 C                     PUT  INVTRAN             03    WRITE TO TRANS FILE
0179 0134 C           END3      TAG                            END SESSION

     INDICATOR NAME USAGES                 RIGHT * = RECORD TYPE REFERENCE
       NAME       STMT#                    LEFT  * = DEFINITION
       EW         *0077  *0113  *0144  *0171
       KG          0077   0078   0113   0114   0144   0145   0171   0172
       KH          0115   0146   0173
       01         *0024  *0033  *0053  *0083   0084   0089   0090   0123*  0123
       02         *0003  *0085   0086   0089   0090   0152*  0152
       03         *0014  *0087   0088   0089   0090   0178*  0178
       72         *0119   0120   0121
```

Figure 11-3 (Part 6 of 13). Generation Printout for Example Program 1

```
                                                                                    WSULIB   IM110W


INDICATOR NAME USAGES                 RIGHT * = RECORD TYPE REFERENCE
  NAME       STMT#                    LEFT  * = DEFINITION
  89         *0032  *0052  *0079  *0080   0081   0082


INDICATORS USED
  EW KG KH 01 02 03 72 89


FIELD NAME AND LABEL USAGES           RIGHT * = UPDATE REFERENCE
  NAME       STMT#                    LEFT  * = DEFINITION
  *ERROR
  *RLNO
  *RLRN
  *RLRR
  *RLRU
  *SLNO
  *USID
  *WSID
  ACODE      *0027*  0101*  0119
  ARCOD      *0034  *0054
  AVAIL      *0039
  BUFFR      *0051  *0066
  CUSNO      *0007*  0138*
  END         0078  *0091
  END1        0114  *0124
  END2        0145  *0153
  END3        0172  *0179
  INVNO      *0008*  0136*
  ITCLS      *0016* *0058
  ITDSC      *0013* *0023* *0031* *0056   0097   0130   0159
  ITNBR      *0005* *0017* *0026*  0032  *0035   0052  *0055   0075*  0095   0128   0157
  ITYPE      *0057
  MDATE      *0048  *0063
  ONORD      *0038
  PONBR      *0019*  0161*
  PRICE      *0010* *0018* *0062   0132   0150
  QADMO      *0044
  QADYR      *0047
  QRCMO      *0043
  QRCYR      *0046
  QTYIS      *0028*  0140*  0150
  QTYOR      *0006*
  QTYRC      *0020*  0099*  0163*
  QUSMO      *0042
  QUSYR      *0045
  RCODE      *0004* *0015* *0025*  0070*  0083   0085   0087
  RENUM      *0041
  REORD      *0040
  RRFST      *0049  *0064
  RRLST      *0050  *0065
  STOCK      *0037
```

Figure 11-3 (Part 7 of 13). Generation Printout for Example Program 1



```
                                                                                           WSULIB   IM110W

       FIELD NAME AND LABEL USAGES                  RIGHT  * = UPDATE REFERENCE
         NAME       STMT#                           LEFT   * = DEFINITION
         TCOST      *0011*  0150*
         TDATE      *0009* *0021* *0029*  0122*  0151*  0177*
         UCOST      *0061   0165
         UDATE       0122   0151   0177
         UDAY
         UMONTH
         UYEAR
         WEGHT      *0060
         WHSLC      *0059
         WRHSE      *0012* *0022* *0030* *0036   0103   0134   0167

       MODE LEVEL DATA FIELD NAMES USED
         NAME   DIM  STMT#   LNG    DEC    DISP  LCL F/ICNT SCR CHN
         RCODE       0004    001     A     0000        003   D
         ITNBR       0005    008     A     0001        004   D   M
         QTYOR       0006    006     0     0009        001
         CUSNO       0007    008     A     000F        001   D
         INVNO       0008    008     A     0017        001   D
         TDATE       0009    006     A     001F        003
         PRICE       0010    009     2     0025        002   D
         TCOST       0011    009     2     002E        001
         WRHSE       0012    001     A     0037        004   D
         ITDSC       0013    030     A     0038        003   D
         ITCLS       0016    002     A     0056        001
         PONBR       0019    007     0     0058        001   D
         QTYRC       0020    006     0     005F        001   D
         ACODE       0027    001     A     0065        001   D
         QTYIS       0028    006     0     0066        001   D
         UCOST       0061    009     2     006C        000   D

       JOB LEVEL DATA FIELD NAMES USED
         NAME   DIM  STMT#   LNG    DEC    DISP  LCL F/ICNT SCR CHN
         UDATE       RSVD    006     0     009A        000

       PROGRAM LABEL NAMES USED
         NAME       STMT#    TYPE
         END        0091     TAG
         END1       0124     TAG
         END2       0153     TAG
         END3       0179     TAG

WSU-0325 UNREFERENCED FIELD NAMES
         NAME       STMT#
         ARCOD      0034
         STOCK      0037
         ONORD      0038
         AVAIL      0039
         REORD      0040
         RENUM      0041
         QUSMO      0042
```

Figure 11-3 (Part 8 of 13). Generation Printout for Example Program 1

```
                                                                                           WSULIB   IM110W

WSU-0325 UNREFERENCED FIELD NAMES
         NAME       STMT#
         QRCMO      0043
         QADMO      0044
         QUSYR      0045
         QRCYR      0046
         QADYR      0047
         MDATE      0048
         RRFST      0049
         RRLST      0050
         BUFFR      0051
         ARCOD      0054
         ITYPE      0057
         WHSLC      0059
         WEGHT      0060
         MDATE      0063
         RRFST      0064
         RRLST      0065
         BUFFR      0066

       SOURCE/OBJECT CROSS REFERENCES
         STMT#    CSB  IAR  SSS     ****** DISPLAYED TEXT FOR PROGRAMMED HALTS (IMSG/MSG/DEBUG) ******
         0081     0000 0025 0009     ENTER VALID ITEM NUMBER
         0089     0000 0063 0009     INVALID FUNCTION
         0120     0007 003B 000A     INVALID CODE

       MAIN STORAGE REQUIREMENTS FOR WSU PROGRAM EXECUTION         MINIMUM        MAXIMUM
        BYTES REQUIRED PER WORK STATION                                 28            284
        TIMES MAXIMUM NUMBER OF WORK STATIONS                            4              4
        PLUS ADDITIONAL STORAGE REQUIRED                              9824          47368
        YIELDS STORAGE REQUIRED FOR EXECUTION                         9936          48504

        REGION REQUIRED FOR EXECUTION                                  10K            48K

       DISK STORAGE REQUIREMENTS FOR WSU EXECUTION PROGRAM WORKFILE
        SECTORS REQUIRED PER WORK STATION                               12
        TIMES MAXIMUM NUMBER OF WORK STATIONS                            4
        PLUS ADDITIONAL DISK SECTORS REQUIRED                            5
        YIELDS MINIMUM DISK SECTORS REQUIRED                            53

        EXECUTION WORKFILE SIZE IN BLOCKS                                6

       IM110W   PROCEDURE CREATED FOR EXECUTION
        // ATTR NEP-NO,MRTMAX-4
        // REGION SIZE-10
        // LOAD #WSXI1
        // FILE NAME-INVTRAN,
        // IF DATAF1-INVTRAN DISP-OLD
        // ELSE DISP-NEW,RECORDS-100
```

Figure 11-3 (Part 9 of 13). Generation Printout for Example Program 1

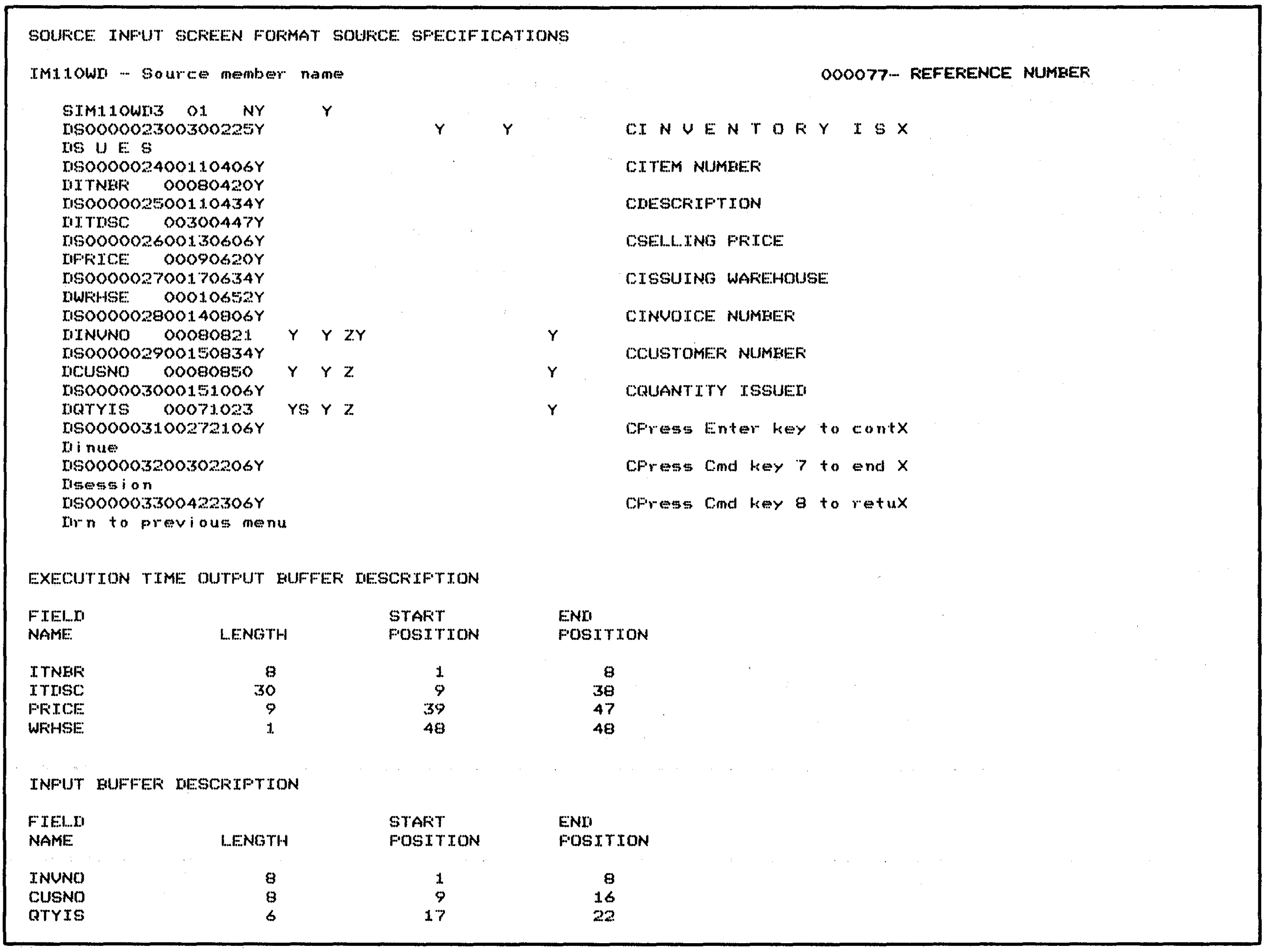
```
                                                                                           WSULIB    IM110W
        // FILE NAME-ITEMBALN,DISP-OLD
        // FILE NAME-ITEMMSTR,DISP-OLD
        // RUN
        // WSX OBJLIBR-WSULIB,OBJMBR-IM110W,FMTLIBR-WSULIB,FMTMBR-IM110WD
        // END

       DIAGNOSTIC TEXT
NOTE#   SEV        MESSAGE TEXT
WSU-0325  W  LISTED FIELD NAME DEFINED BUT NOT REFERENCED.

                  GENERATION SUCCESSFUL---WARNING ERRORS LISTED ABOVE.
```

```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM110WD - Source member name                                          000077- REFERENCE NUMBER

   SIM110WD1  01     NY     Y
   DS00000001006802O5Y                 Y     Y          CI N V E N T O R Y  A D X
   DJ U S T M E N T S/I S S U E S/R E C E I P T S
   DS00000002001505 27Y                       Y          CSELECT FUNCTION
   DRCODE     00010544   YA Y  Y      Y         Y
   DS0000000300140730Y                                  CADJUSTMENTS -A
   DS0000000400140830Y                                  CISSUES      -I
   DS0000000500140930Y                                  CRECEIPTS    -R
   DS0000000600111127Y                                  CITEM NUMBER
   DITNBR     00081139   YN Y B                 Y
   DS0000000700302206Y                                  CPress Cmd key 7 to end X
   Dsession

INPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

RCODE                1              1              1
ITNBR                8              2              9
```

Figure 11-3 (Part 10 of 13). Generation Printout for Example Program 1

```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM110WD - Source member name                                          000077- REFERENCE NUMBER

   SIM110WD2  01  YNY     Y
   DS000000800400221Y               Y     Y          CI N V E N T O R Y  A D X
   DJ U S T M E N T S
   DS000000900110406Y                                CITEM NUMBER
   DITNBR    00080419Y
   DS000001000110434Y                                CDESCRIPTION
   DITDSC    00300447Y
   DS000001100170606Y                                CQUANTITY RETURNED
   DQTYRC    00070624   YS Y ZY                Y
   DS000001200150634Y                                CADJUSTMENT CODE
   DACODE    00010650   YN Y                   Y
   DS000001300190806Y                                CRECEIVING WAREHOUSE
   DWRHSE    00010826Y
   DS000001400220834Y                                CSELECT ADJUSTMENT CODE
   DS000001500220936Y                                C1- Damaged in shipping
   DS000001600171036Y                                C2- Incorrect Size
   DS000001700181136Y                                C3- Incorrect Color
   DS000001800171236Y                                C4- Incorrect Item
   DS000001900081336Y                                C5- Other
   DS000002000272106Y                                CPress Enter key to contX
   Dinue
   DS000002100302206Y                                CPress Cmd key 7 to end X
   Dsession
   DS000002200422306Y                                CPress Cmd key 8 to retuX
   Drn to previous menu

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

ITNBR                8             1              8
ITDSC               30             9             38
WRHSE                1            39             39

INPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

QTYRC                6             1              6
ACODE                1             7              7
```

Figure 11-3 (Part 11 of 13). Generation Printout for Example Program 1



```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM110WD - Source member name                                           000077- REFERENCE NUMBER

   SIM110WD3  01    NY      Y
   DS000002300300225Y                Y     Y         CI N V E N T O R Y  I S X
   DS U E S
   DS000002400110406Y                                CITEM NUMBER
   DITNBR   00080420Y
   DS000002500110434Y                                CDESCRIPTION
   DITDSC   00300447Y
   DS000002600130606Y                                CSELLING PRICE
   DPRICE   00090620Y
   DS000002700170634Y                                CISSUING WAREHOUSE
   DWRHSE   00010652Y
   DS000002800140806Y                                CINVOICE NUMBER
   DINVNO   00080821   Y  Y ZY                 Y
   DS000002900150834Y                                CCUSTOMER NUMBER
   DCUSNO   00080850   Y  Y Z                  Y
   DS000003000151006Y                                CQUANTITY ISSUED
   DQTYIS   00071023   YS Y Z                  Y
   DS000003100272106Y                                CPress Enter key to contX
   Dinue
   DS000003200302206Y                                CPress Cmd key 7 to end X
   Dsession
   DS000003300422306Y                                CPress Cmd key 8 to retuX
   Drn to previous menu

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

ITNBR                8              1              8
ITDSC               30              9             38
PRICE                9             39             47
WRHSE                1             48             48

INPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

INVNO                8              1              8
CUSNO                8              9             16
QTYIS                6             17             22
```

Figure 11-3 (Part 12 of 13). Generation Printout for Example Program 1

```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM110WD - Source member name                                          000077- REFERENCE NUMBER

   SIM110WD4  01    NY      Y
   DS000003400340221Y                Y     Y         CI N V E N T O R Y  R E X
   DC E I P T S
   DS000003500110406Y                                CITEM NUMBER
   DITNBR    00080419Y
   DS000003600110434Y                                CDESCRIPTION
   DITDSC    00300447Y
   DS000003700120606Y                                CORDER NUMBER
   DPONBR    00080619   YS Y ZY                Y
   DS000003800170634Y                                CQUANTITY RECEIVED
   DQTYRC    00070653   YS Y Z                 Y
   DS000003900090806Y                                CITEM COST
   DUCOST    00090816Y
   DS000004000190834Y                                CRECEIVING WAREHOUSE
   DWRHSE    00010856Y                         Y
   DS000004100272106Y                                CPress Enter key to contX
   Dinue
   DS000004200302206Y                                CPress Cmd key 7 to end X
   Dsession
   DS000004300422306Y                                CPress Cmd key 8 to retuX
   Drn to previous menu

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

ITNBR                8             1              8
ITDSC               30             9             38
UCOST                9            39             47
WRHSE                1            48             48

INPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

PONBR                7             1              7
QTYRC                6             8             13
```

```
  WSULIB   - INPUT LIBRARY NAME

  WSULIB   - Output library name

  IM110WD  - Format load member name                                   000077- REFERENCE NUMBER

FORMAT IM110WD1 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF    223 BYTES
FORMAT IM110WD2 REQUIRES   768 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF    443 BYTES
FORMAT IM110WD3 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF    361 BYTES
FORMAT IM110WD4 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF    338 BYTES
```

Figure 11-3 (Part 13 of 13). Generation Printout for Example Program 1

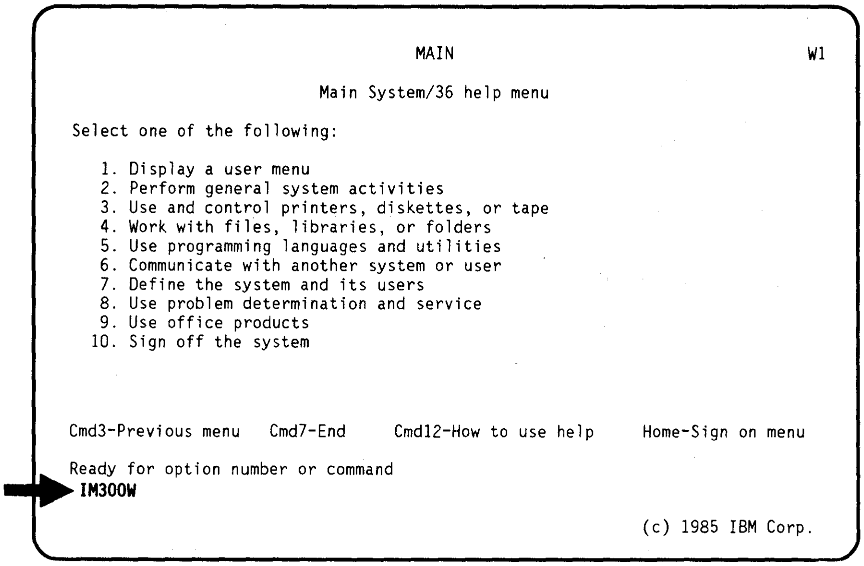
Example Program 2: Item Master File Maintenance

This program, named IM300W, allows a maximum of two operators to concurrently enter additions, changes, and deletions to a master file (ITEMMSTR). Input from operators consists of entering an item number, selecting a function, and responding to requests from the display.

Output consists of a master file that contains all operator entered additions, changes, and deletions (ITEMMSTR).

Running IM300W

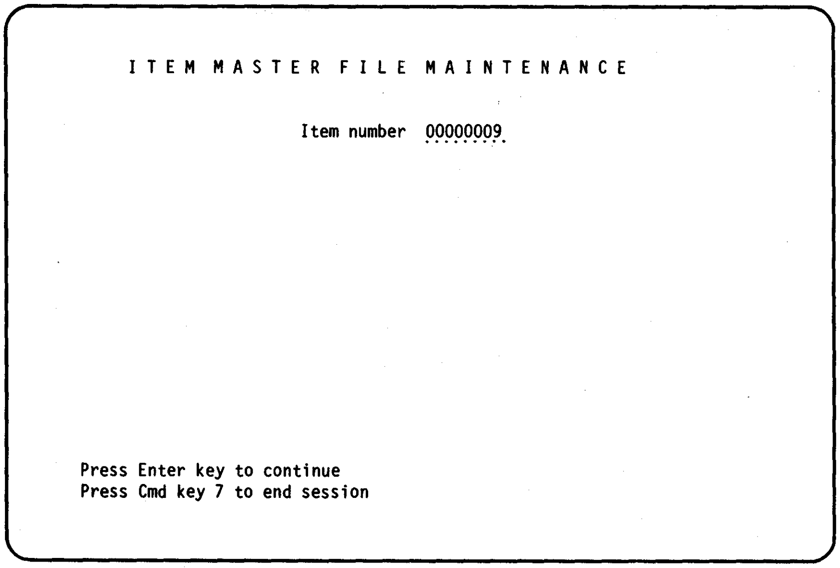
To initiate WSU program IM300W, the operator enters the procedure name, IM300W.



```
                                MAIN                                    W1

                        Main System/36 help menu

  Select one of the following:

     1. Display a user menu
     2. Perform general system activities
     3. Use and control printers, diskettes, or tape
     4. Work with files, libraries, or folders
     5. Use programming languages and utilities
     6. Communicate with another system or user
     7. Define the system and its users
     8. Use problem determination and service
     9. Use office products
    10. Sign off the system



  Cmd3-Previous menu   Cmd7-End     Cmd12-How to use help     Home-Sign on menu

  Ready for option number or command
   IM300W
                                                           (c) 1985 IBM Corp.
```

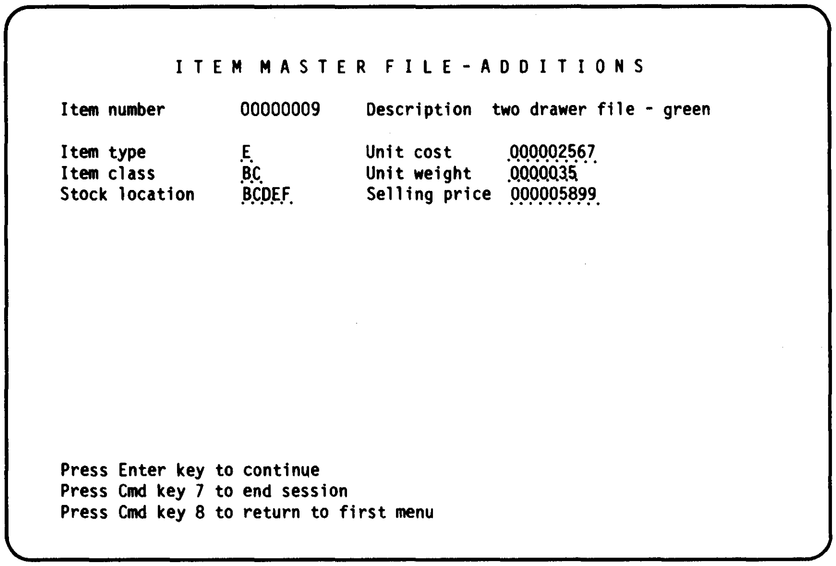
This display occurs when an operator initiates the WSU program IM300W.

The operator enters either a new item number or an existing item number.



```
         I T E M   M A S T E R   F I L E   M A I N T E N A N C E

                            Item number  00000009










      Press Enter key to continue
      Press Cmd key 7 to end session
```

If the operator enters a new item number, the title of the next display is Item Master File - Additions.



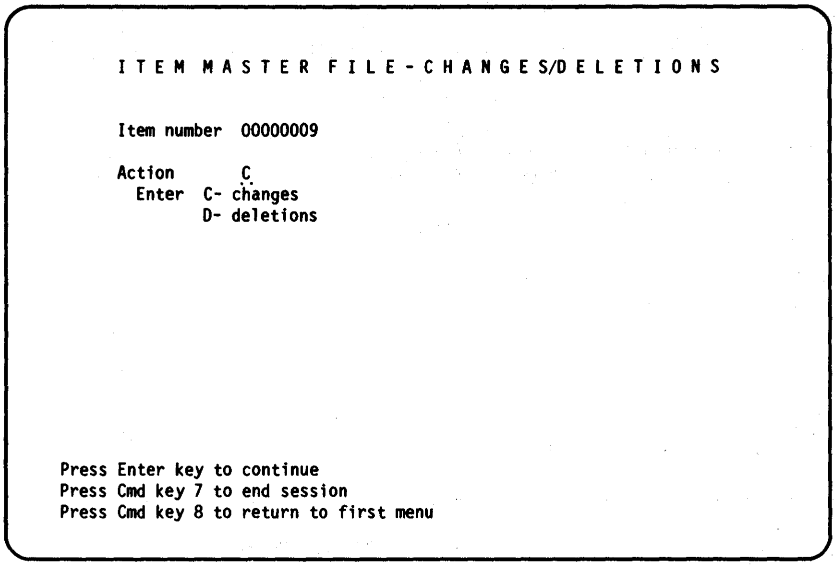
```
            I T E M   M A S T E R   F I L E - A D D I T I O N S

      Item number       00000009     Description  two drawer file - green

      Item type         E            Unit cost     000002567
      Item class        BC           Unit weight   0000035
      Stock location    BCDEF        Selling price 000005899













      Press Enter key to continue
      Press Cmd key 7 to end session
      Press Cmd key 8 to return to first menu
```

This display has seven input fields:

- Description
- Item type
- Unit cost
- Item class
- Unit weight
- Stock location
- Selling price.

An entry is required in all fields. Command key 8 causes the previous display to be shown without any processing on the additions display.

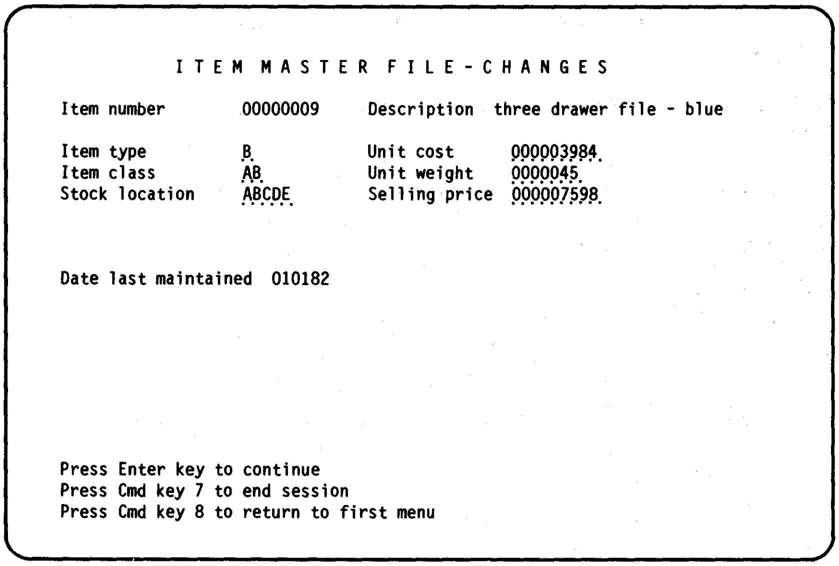
If the operator entered an existing item number on the first display, the title of this display is Item Master File - Changes/Deletions.



```
      ITEM MASTER FILE-CHANGES/DELETIONS

      Item number  00000009

      Action        C
        Enter  C- changes
               D- deletions









Press Enter key to continue
Press Cmd key 7 to end session
Press Cmd key 8 to return to first menu
```

This display has one input field (Action). Command key 8 causes the previous display to be shown without any processing.

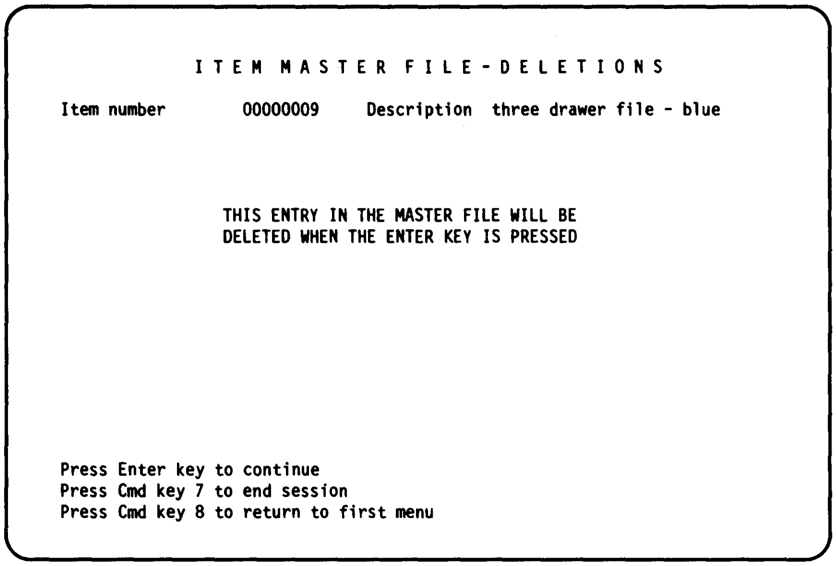
If the operator selects action C (changes) on this display, the title of the next display is Item Master File - Changes.



```
            ITEM MASTER FILE-CHANGES

Item number       00000009     Description  three drawer file - blue

Item type         B            Unit cost     000003984
Item class        AB           Unit weight   0000045
Stock location    ABCDE        Selling price 000007598



Date last maintained  010182









Press Enter key to continue
Press Cmd key 7 to end session
Press Cmd key 8 to return to first menu
```

When this display appears, WSU shows all of the current information about this item from the master file. All of the fields except the item number and the date of the last change can be changed by the operator. When the Enter key is pressed, processing continues and the first display is shown again. Command key 8 causes the previous display to be shown again without any processing.

If the operator enters a D on the Changes/Deletions display, the Item Master File - Deletions display appears.



```
                ITEM MASTER FILE-DELETIONS

Item number        00000009     Description  three drawer file - blue



                   THIS ENTRY IN THE MASTER FILE WILL BE
                   DELETED WHEN THE ENTER KEY IS PRESSED









Press Enter key to continue
Press Cmd key 7 to end session
Press Cmd key 8 to return to first menu
```

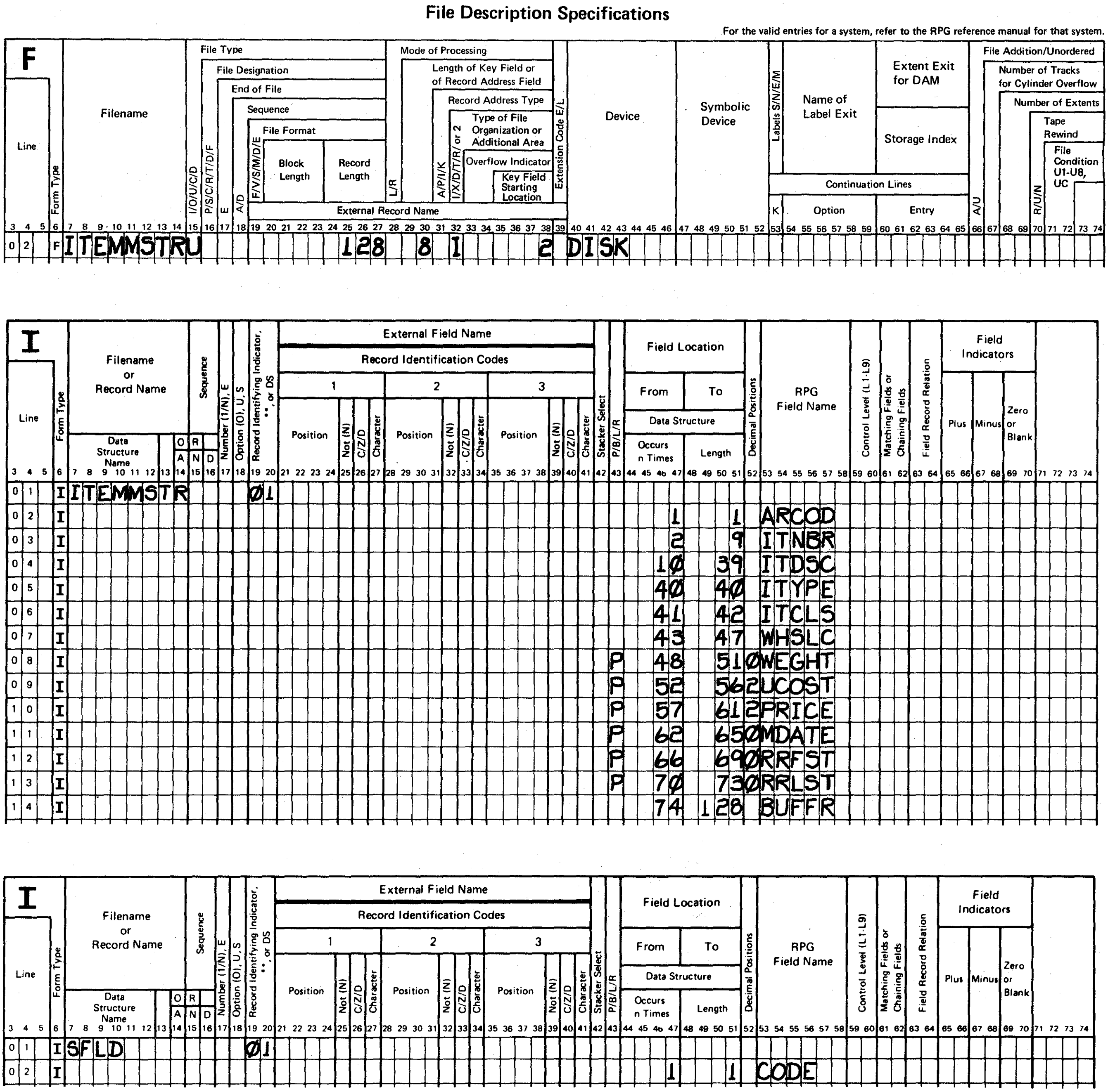
No entries are allowed on this display. When the operator presses the Enter key, the current item number is marked for deletion and cannot be called up for changes again. Command key 8 causes the previous display to be shown again without any processing.

Coding Forms

In this example, the file definition uses one source member and one session-level field:

- FILESRC contains F- and I-specifications for the item master file, ITEMMSTR.
- SFLD is the session-level field used for the change code.

Figure 11-4 shows the file definition for example program 2.



File Description Specifications

For the valid entries for a system, refer to the RPG reference manual for that system.

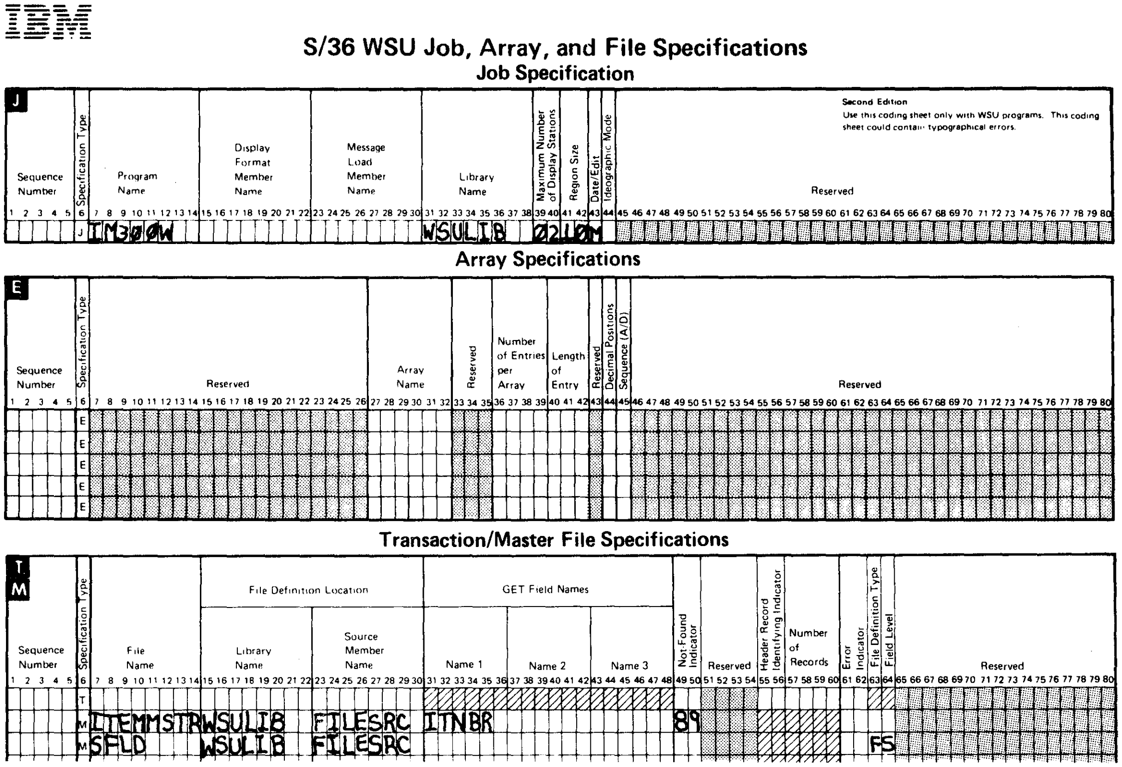
```
02 FITEMMSTRU         128  8 I     2 DISK
```

```
01 IITEMMSTR    01
02 I                                        1   1 ARCOD
03 I                                        2   9 ITNBR
04 I                                       10  39 ITDSC
05 I                                       40  40 ITYPE
06 I                                       41  42 ITCLS
07 I                                       43  47 WHSLC
08 I                                    P  48  510WEGHT
09 I                                    P  52  562UCOST
10 I                                    P  57  612PRICE
11 I                                    P  62  650MDATE
12 I                                    P  66  690RRFST
13 I                                    P  70  730RRLST
14 I                                       74 128 BUFFR
```

```
01 ISFLD        01
02 I                                        1   1 CODE
```

Figure 11-4. File Definition for Example Program 2

The source program is in a source member named IM300W. Figure 11-5 shows the WSU specifications for IM300W.



IBM

S/36 WSU Job, Array, and File Specifications

Job Specification

Second Edition
Use this coding sheet only with WSU programs. This coding sheet could contain typographical errors.

J Sequence Number 1 2 3 4 5	Specification Type 6	Program Name 7 8 9 10 11 12 13 14	Display Format Member Name 15 16 17 18 19 20 21 22	Message Load Member Name 23 24 25 26 27 28 29 30	Library Name 31 32 33 34 35 36 37 38	Maximum Number of Display Stations 39 40	Region Size 41 42	Date/Edit 43	Ideographic Mode 44	Reserved 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	J	IM300W			WSULIB	02	10	M		

Array Specifications

E Sequence Number 1 2 3 4 5	Specification Type 6	Reserved 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	Array Name 27 28 29 30 31 32	Reserved 33 34 35	Number of Entries per Array 36 37 38 39	Length of Entry 40 41 42	Reserved 43	Decimal Positions 44	Sequence (A/D) 45	Reserved 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	E									
	E									
	E									
	E									
	E									

Transaction/Master File Specifications

T M			File Definition Location		GET Field Names										
Sequence Number 1 2 3 4 5	Specification Type 6	File Name 7 8 9 10 11 12 13 14	Library Name 15 16 17 18 19 20 21 22	Source Member Name 23 24 25 26 27 28 29 30	Name 1 31 32 33 34 35 36	Name 2 37 38 39 40 41 42	Name 3 43 44 45 46 47 48	Not-Found Indicator 49 50	Reserved 51 52 53 54	Header Record Identifying Indicator 55 56	Number of Records 57 58 59 60	Error Indicator 61 62	File Definition Type 63	Field Level 64	Reserved 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	T														
	M	ITEMMSTR	WSULIB	FILESRC	ITNBR			89							
	M	SFLD	WSULIB	FILESRC									F	S	

Figure 11-5 (Part 1 of 7). WSU Specifications for Example Program 2

IBM

System/36 Display Format Specifications

Display Control Specification

S — Sequence Number (1–5); Specification Type (6); Display Format Name (7–14); Format ID (WSU Only) (15–16); Start Line Number (17–18); Number of Lines to Clear (19–20); Lowercase (21); Return Input (22); Reset Keyboard (WSU Only) (23–24); Sound Alarm (25–26); Enable Function Keys (27); Enable Command Keys (28); Blink Cursor (29–30); Erase Input Fields (31–32); Override Fields (33–34); Suppress Input (35–36); Null Fill (37–38); Reserved (39); Right-to-Left Display (40); WSU Only: Enter Mode Sequence: Start (41), End (42), Entry Required (43), Repeat (44); Reserved (45); Priority (46); Preprocess (47); Review Mode Record Identifying Indicators 1 2 3 (48–53); Insert Mode Record Identifying Indicators 1 2 3 (54–59); Reserved (60–63); Key Mask (64–79); Reserved (80)

```
     SIM300WD1              Y           Y Y
```

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

D — Sequence Number (1–5); Specification Type (6); WSU Field Name / Field Name (7–14); WSU Arrays: Elements per Row or Column (15–16), Fill Direction (17), Line Skip Factor (18), Horizontal Positions Space Factor / Field Length (15–18); Starting Location: Line Number (19–20), Horizontal Position (21–22); Output Data (23–24); Edit Code (WSU Only) (25); Input Allowed (26); Data Type (27); Mandatory Fill (28); Mandatory Entry (29); Self-Check (30); Adjust/Fill (31); Position Cursor (32–33); Enable Dup (34); Controlled Field Exit (35); Auto Record Advance (36); Protect Field (37–38); High Intensity (39–40); Blink Field (41–42); Nondisplay (43–44); Reverse Image (45–46); Underline (47–48); Column Separators (49); Reserved (50); Lowercase (51); Reserved (52–55); Constant Type (56); Constant Data (57–79); Continuation (80)

```
     D            0213                Y     Y          P'I T E M  M A S T E R  X
     DF I L E  M A I N T E N A N C E'
     D            0531                                 P'Item number'
     DITNBR       0544   YN Y Z                 Y
     D            2106                                 P'Press Enter key to conX
     Dtinue'
     D            2206                                 P'Press Cmd key 7 to endX
     D session'
```

S/36 WSU Processing Specifications

C — Sequence Number (1–5); Specification Type (6); Processing Function (7–8); Conditioning Indicators: Not (9), (10–11), And: Not (12), (13–14), And: Not (15), (16–17); Factor 1 (18–27); Operation (28–32); Table or Message Text: Factor 2 (33–42), Literal / Result Field: Name (43–48), Length (49–51), Decimal Pos (52), Half Adjust (H) (53); Blank / Continuation; Resulting Indicators: Arithmetic: Plus, Minus, Zero; Compare: High 1>2, Low 1<2, Equal 1=2 (54–59); Comments (60–80)

```
     C   KG                SETONEW
     C   KG                GOTO END
     C                     GET  ITEMMSTR           89    ITEM MASTER FILE
     C   89                PUTS IM300WD4                 ADDITIONS TO MASTER
     C           ITNBR     COMP ' '                    69NUMBER NOT VALID
     C   69                MSG  'INVALID ITEM NUMBERX
     C'
     C   69                PUTS IM300WD1                 REPEAT DISPLAY
     C           ARCOD     COMP 'X'                    59DELETED ITEM NUMBER
     C   59                MSG  'DELETED ITEM'
     C   59                PUTS IM300WD1                 REPEAT DISPLAY
     C           END       TAG                           END SESSION
```

Figure 11-5 (Part 2 of 7). WSU Specifications for Example 2

IBM

System/36 Display Format Specifications

First Edition

Use this coding sheet only to define display formats for WSU and $SFGR. This coding sheet could contain typographical errors.

Display Control Specification

S

Sequence Number	Specification Type	Display Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Null Fill	Reserved	Right-to-Left Display	WSU Only: Enter Mode Sequence Start	End	Entry Required	Repeat	Reserved	Reserved	Priority	Preprocess	Review Mode Record Identifying Indicators 1	2	3	Insert Mode Record Identifying Indicators 1	2	3	Reserved	Key Mask	Reserved
	S	IM300WD2																			Y														

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

D

Sequence Number	Specification Type	WSU Field Name / Field Name	WSU Arrays: Elements per Row or Column, Fill Direction, Line Skip Factor, Horizontal Positions Space Factor / Field Length	Starting Location Line Number	Horizontal Position	Output Data	Edit Code (WSU Only)	Input Allowed	Data Type	Mandatory Fill	Mandatory Entry	Self-Check	Adjust/Fill	Position Cursor	Enable Dup	Controlled Field Exit	Auto Record Advance	Protect Field	High Intensity	Blink Field	Nondisplay	Reverse Image	Underline	Column Separators	Reserved	Lowercase	Reserved	Constant Type	Constant Data	Continuation
	D			02	10														Y			Y						P	'I T E M M A S T E R	X
	D	F I L E - C H A N G E S/D E L E T I O N S'																												
	D			05	10																							P	'Item number'	
	D	ITNBR		05	23	Y																								
	D			07	10																							P	'Action'	
	D	CODE		07	23			Y	A		Y													Y						
	D			08	12																							P	'Enter C- changes'	
	D			09	19																							P	'D- deletions'	
	D			21	04																							P	'Press Enter key to con	X
	D	tinue'																												
	D			22	04																							P	'Press Cmd key 7 to end	X
	D	session'																												
	D			23	04																							P	'Press Cmd key 8 to ret	X
	D	urn to first menu'																												

S/36 WSU Processing Specifications

C

Sequence Number	Specification Type	Processing Function	Conditioning Indicators Not	Conditioning Indicators	And Not	And	And Not	And	Factor 1	Operation	Factor 2 / Table or Message Text / Literal	Result Field Name	Length	Decimal Pos	Half Adjust (H)	Resulting Indicators Arithmetic Plus / Compare High 1>2	Minus / Low 1<2	Zero / Equal 1=2	Comments
	C			KG						SETON						EW			USE CMD 7 TO END
	C			KG						GOTO	END1								
	C			KH						PUTS	IM300WD1								RETURN TO FIRST
	C								CODE	COMP	'C'							11	CHANGES
	C			11						PUTS	IM300WD3								GET CHANGES DISPLAY
	C								CODE	COMP	'D'							12	DELETIONS
	C			12						PUTS	IM300WD5								GET DELETIONS DISPLAY
	C		N	11	N	12				MSG	'INVALID SELECTION'								INCORRET SELECTION
	C	*																	
	C								END1	TAG									SESSION ENDED

Figure 11-5 (Part 3 of 7). WSU Specifications for Example 2

System/36 Display Format Specifications

Display Control Specification

Sequence Number (1–5)	Specification Type (6)	Display Format Name (7–14)	Format ID (WSU Only) (15–16)	Start Line Number (17–18)	Number of Lines to Clear (19–20)	Lowercase (21)	Return Input (22)	Reset Keyboard (WSU Only) (23–24)	Sound Alarm (25–26)	Enable Function Keys (27)	Enable Command Keys (28)	Blink Cursor (29–30)	Erase Input Fields (31–32)	Override Fields (33–34)	Suppress Input (35–36)	Null Fill (37–38)	Reserved (39)	Right-to-Left Display (40)	WSU Only: Enter Mode Sequence: Start (41)	End (42)	Entry Required (43)	Repeat (44)	Reserved (45)	Priority (46)	Preprocess (47)	Review Mode Record Identifying Indicators 1 (48–49)	2 (50–51)	3 (52–53)	Insert Mode Record Identifying Indicators 1 (54–55)	2 (56–57)	3 (58–59)	Reserved (60–63)	Key Mask (64–79)	Reserved (80)
	S	IM300WD3										Y								Y	Y													

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

Sequence Number (1–5)	Specification Type (6)	WSU Field Name / Field Name (7–14)	WSU Arrays: Elements per Row or Column	Fill Direction	Line Skip Factor	Horizontal Positions Space Factor	Field Length (15–18)	Starting Location: Line Number (19–20)	Horizontal Position (21–22)	Output Data (23–24)	Edit Code (WSU Only) (25)	Input Allowed (26)	Data Type (27)	Mandatory Fill (28)	Mandatory Entry (29)	Self-Check (30)	Adjust/Fill (31)	Position Cursor (32–33)	Enable Dup (34)	Controlled Field Exit (35)	Auto Record Advance (36)	Protect Field (37–38)	High Intensity (39–40)	Blink Field (41–42)	Nondisplay (43–44)	Reverse Image (45–46)	Underline (47–48)	Column Separators (49)	Reserved (50)	Lowercase (51)	Reserved (52–55)	Constant Type (56)	Constant Data (57–79)	Continuation (80)
	D							02	16														Y			Y						P	'I T E M M A S T E R	X
	D	F I L E - C H A N G E S'																																
	D							04	04																							P	'Item number'	
	D	ITNBR						04	23	Y																								
	D							04	36																							P	'Description'	
	D	ITDSC						04	49	Y		Y	B														Y			Y				
	D							06	04																							P	'Item type'	
	D	ITYPE						06	23	Y		Y	B															Y		N				
	D							06	36																							P	'Unit cost'	
	D	UCOST						06	51	Y		Y	N				Z											Y						
	D							07	04																							P	'Item class'	
	D	ITCLS						07	23	Y		Y	B		Y													Y		N				
	D							07	36																							P	'Unit weight'	
	D	WEGHT						07	51	Y		Y	N				Z											Y						
	D							08	04																							P	'Stock location'	

Field Definition Specification

Sequence Number (1–5)	Specification Type (6)	WSU Field Name / Field Name (7–14)	WSU Arrays: Elements per Row or Column	Fill Direction	Line Skip Factor	Horizontal Positions Space Factor	Field Length (15–18)	Starting Location: Line Number (19–20)	Horizontal Position (21–22)	Output Data (23–24)	Edit Code (WSU Only) (25)	Input Allowed (26)	Data Type (27)	Mandatory Fill (28)	Mandatory Entry (29)	Self-Check (30)	Adjust/Fill (31)	Position Cursor (32–33)	Enable Dup (34)	Controlled Field Exit (35)	Auto Record Advance (36)	Protect Field (37–38)	High Intensity (39–40)	Blink Field (41–42)	Nondisplay (43–44)	Reverse Image (45–46)	Underline (47–48)	Column Separators (49)	Reserved (50)	Lowercase (51)	Reserved (52–55)	Constant Type (56)	Constant Data (57–79)	Continuation (80)
	D	WHSLC						08	23	Y		Y	B		Y													Y		N				
	D							08	36																							P	'Selling price'	
	D	PRICE						08	51	Y		Y	N				Z											Y						
	D							12	04																							P	'Date Last Maintained'	
	D	MDATE						12	26	Y																								
	D							21	04																							P	'Press Enter key to con	X
	D	tinue'																																
	D							22	04																							P	'Press Cmd key 7 to end	X
	D	session'																																
	D							23	04																							P	'Press Cmd key 8 to ret	X
	D	urn to first menu'																																

Figure 11-5 (Part 4 of 7). WSU Specifications for Example 2

S/36 WSU Processing Specifications

Sequence Number (1-5)	Specification Type (6)	Processing Function (7-8)	Conditioning Indicators: Not (9)	Conditioning Indicators (10-11)	And: Not (12)	And (13-14)	And: Not (15)	And (16-17)	Factor 1 (18-27)	Operation (28-32)	Table or Message Text / Factor 2 (33-42)	Literal / Result Field: Name (43-48)	Length (49-51)	Decimal Pos (52)	Half Adjust (H) (53)	Resulting Indicators: Arithmetic Plus / Compare High 1>2 (54-55)	Minus / Low 1<2 (56-57)	Zero / Equal 1=2 (58-59)	Comments (60-80)
	C			KG						SETON						EW			CMD 7 TO END SESSION
	C			KG						GOTO	END2								
	C			KH						PUTS	IM300WD1								RETURN TO FIRST
	C									MOVE	UDATE	MDATE							DATE OF CHANGES
	C									PUT	ITEMMSTR								CHANGES TO ITEM FILE
	C								END2	TAG									

Figure 11-5 (Part 5 of 7). WSU Specifications for Example 2

IBM

System/36 Display Format Specifications

Display Control Specification

Sequence Number (1-5)	Specification Type (6)	Display Format Name (7-14)	Blink Cursor (29-30)	Entry Required (43)
	S	IM300WD4	Y	Y

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

Sequence Number	Specification Type	Field Name	Line Number	Horizontal Position	Output Data	Input Allowed	Data Type	Mandatory Entry	Adjust/Fill	High Intensity	Reverse Image	Underline	Column Separators	Lowercase	Constant Type	Constant Data	Continuation
	D		02	16						Y	Y				P	'ITEM MASTER	X
	D	FILE - ADDITIONS'															
	D		04	04											P	'Item number'	
	D	ITNBR	04	23	Y												
	D		04	36											P	'Description'	
	D	ITDSC	04	49		Y	B	Y				Y		Y			
	D		06	04											P	'Item type'	
	D	ITYPE	06	23		Y	B	Y					Y	N			
	D		06	36											P	'Unit cost'	
	D	UCOST	06	51		Y	N	Y	Z				Y				
	D		07	04											P	'Item class'	
	D	ITCLS	07	23		Y	B	Y					Y	N			
	D		07	36											P	'Unit weight'	
	D	WEGHT	07	51		Y	N	Y	Z				Y				
	D		08	04											P	'Stock location'	
	D	WHSLC	08	23		Y	B	Y					Y	N			
	D		08	36											P	'Selling price'	
	D	PRICE	08	51		Y	N	Y	Z				Y				
	D		21	04											P	'Press Enter key to con	X
	D	tinue'															
	D		22	04											P	'Press Cmd key 7 to end	X
	D	session'															
	D		23	04											P	'Press Cmd key 8 to ret	X
	D	urn to first menu'															

S/36 WSU Processing Specifications

Sequence Number	Specification Type	Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators	Comments
	C	KG		SETON			EW	CMD 7 TO END SESSION
	C	KG		GOTO	END3			
	C	KH		PUTS	IM300WD1			RETURN TO FIRST
	C			MOVE	UDATE	MDATE		DATE FIRST ENTERED
	C			PUTN	ITEMMSTR		01	ITEM MASTER FILE
	C			PUTS	IM300WD1			RETURN TO FIRST
	C		END3	TAG				

Figure 11-5 (Part 6 of 7). WSU Specifications for Example 2

IBM

System/36 Display Format Specifications

Display Control Specification

Sequence Number	Specification Type	Display Format Name	Lowercase
	S	IM300WD5	Y

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

Sequence Number	Specification Type	WSU Field Name	Line Number	Horizontal Position	Output Data	High Intensity	Reverse Image	Constant Type	Constant Data	Continuation
	D		02	18		Y	Y	P	'ITEM MASTER	X
	D	FILE-DELETIONS'								
	D		04	04				P	'Item number'	
	D	ITNBR	04	23	Y					
	D		04	36				P	'Description'	
	D	ITDSC	04	49	Y					
	D		09	21		Y		P	'THIS ENTRY IN THE MAST	X
	D	ER FILE WILL BE'								
	D		10	21		Y		P	'DELETED WHEN THE ENTER	X
	D	KEY IS PRESSED'								
	D		21	04				P	'Press Enter key to con	X
	D	tinue'								
	D		22	04				P	'Press Cmd key 7 to end	X
	D	session'								
	D		23	04				P	'Press Cmd key 8 to ret	X
	D	urn to first menu								

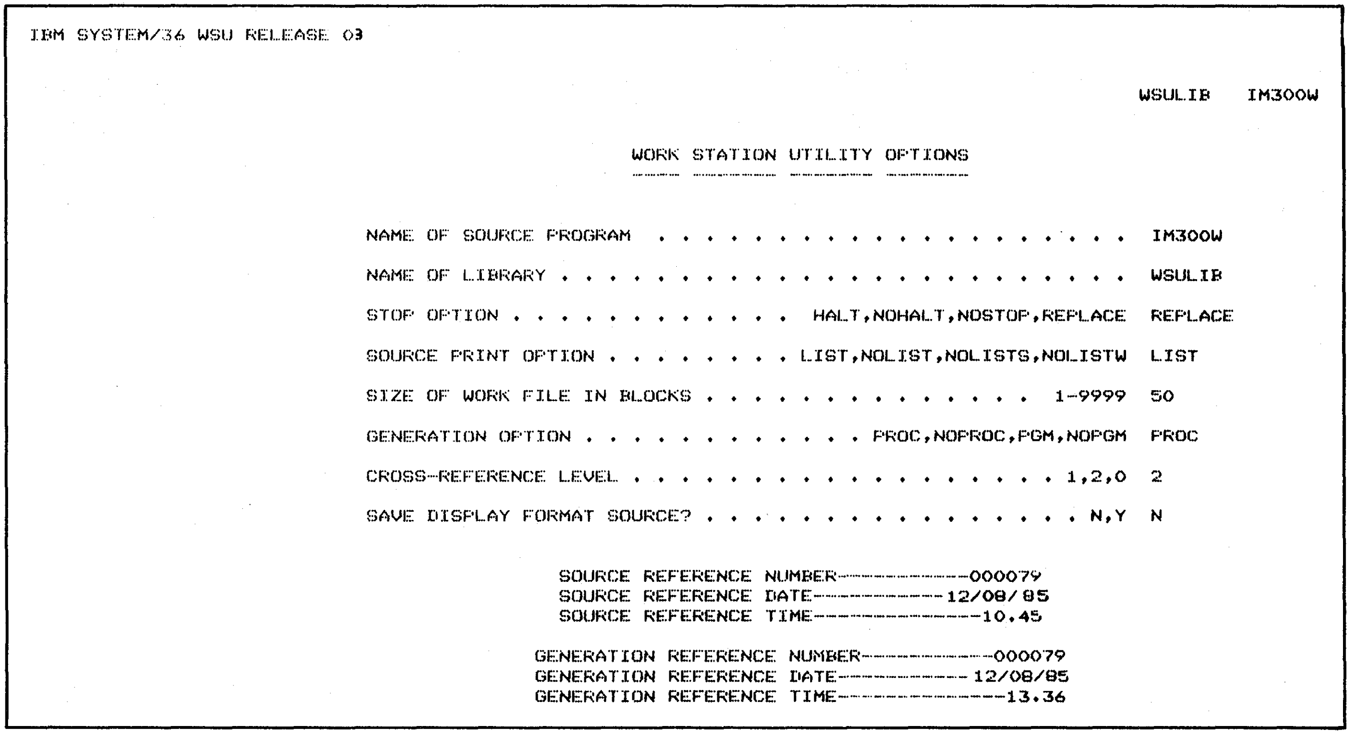
S/36 WSU Processing Specifications

Sequence Number	Specification Type	Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators	Comments
	C	KG		SETON			EW	CMD 7 TO END SESSION
	C	KG		GOTO	END4			
	C	KH		PUTS	IM300WD1			RETURN TO FIRST
	C			MOVE	'X'	ARCOD		DELETE FROM MSTR FILE
	C			MOVE	UDATE	MDATE		WRITE DATE DELETED
	C			PUT	ITEMMSTR			WRITE TO MASTER
	C			PUTS	IM300WD1			RETURN TO FIRST
	C		END4	TAG				SESSION ENDED

Figure 11-5 (Part 7 of 7). WSU Specifications for Example 2

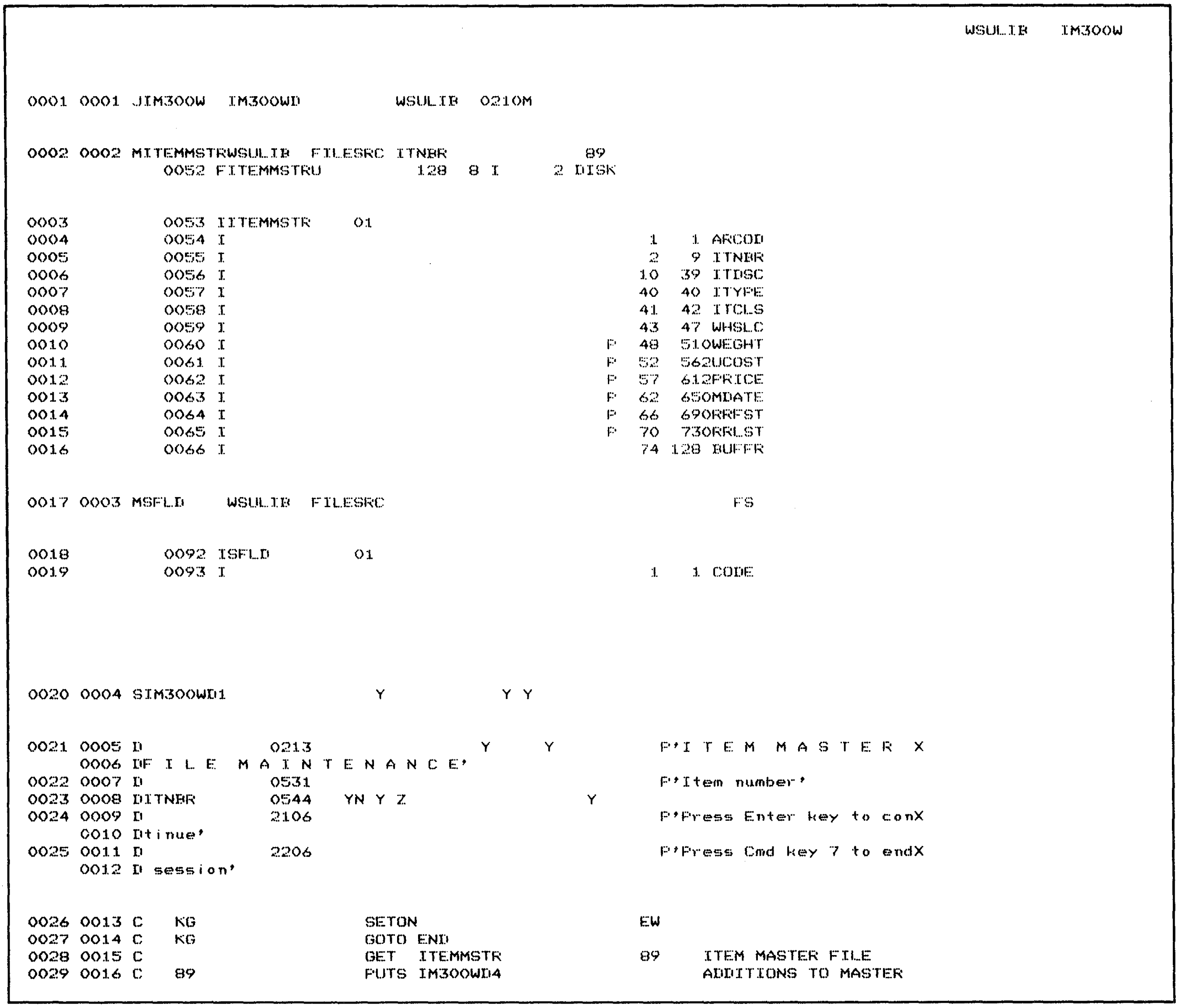
Program Generation Printout

Figure 11-6 shows the generation printout of IM300W.



```
IBM SYSTEM/36 WSU RELEASE 03

                                                                                                  WSULIB    IM300W

                                                        WORK STATION UTILITY OPTIONS
                                                        ---- ------- ------- -------

                                   NAME OF SOURCE PROGRAM  . . . . . . . . . . . . . . . . . . . . . .  IM300W

                                   NAME OF LIBRARY . . . . . . . . . . . . . . . . . . . . . . . . . .  WSULIB

                                   STOP OPTION . . . . . . . . . . . .  HALT,NOHALT,NOSTOP,REPLACE  REPLACE

                                   SOURCE PRINT OPTION . . . . . . . . LIST,NOLIST,NOLISTS,NOLISTW  LIST

                                   SIZE OF WORK FILE IN BLOCKS . . . . . . . . . . . . . . .  1-9999  50

                                   GENERATION OPTION . . . . . . . . . . . . . PROC,NOPROC,PGM,NOPGM  PROC

                                   CROSS-REFERENCE LEVEL . . . . . . . . . . . . . . . . . . . 1,2,0  2

                                   SAVE DISPLAY FORMAT SOURCE? . . . . . . . . . . . . . . . . . N,Y  N

                                                  SOURCE REFERENCE NUMBER--------------000079
                                                  SOURCE REFERENCE DATE-------------12/08/85
                                                  SOURCE REFERENCE TIME----------------10.45

                                                GENERATION REFERENCE NUMBER--------------000079
                                                GENERATION REFERENCE DATE-------------12/08/85
                                                GENERATION REFERENCE TIME----------------13.36
```

Figure 11-6 (Part 1 of 12). Generation Printout for Example Program 2



```
                                                                                         WSULIB   IM300W


0001 0001 JIM300W  IM300WD         WSULIB  0210M


0002 0002 MITEMMSTRWSULIB  FILESRC ITNBR             89
             0052 FITEMMSTRU         128  8 I     2 DISK


0003         0053 IITEMMSTR    01
0004         0054 I                                        1   1 ARCOD
0005         0055 I                                        2   9 ITNBR
0006         0056 I                                       10  39 ITDSC
0007         0057 I                                       40  40 ITYPE
0008         0058 I                                       41  42 ITCLS
0009         0059 I                                       43  47 WHSLC
0010         0060 I                                    P  48  510WEGHT
0011         0061 I                                    P  52  562UCOST
0012         0062 I                                    P  57  612PRICE
0013         0063 I                                    P  62  650MDATE
0014         0064 I                                    P  66  690RRFST
0015         0065 I                                    P  70  730RRLST
0016         0066 I                                       74 128 BUFFR


0017 0003 MSFLD    WSULIB  FILESRC                                 FS


0018         0092 ISFLD        01
0019         0093 I                                        1   1 CODE





0020 0004 SIM300WD1              Y           Y Y


0021 0005 D            0213                Y     Y          P'I T E M   M A S T E R  X
     0006 DF I L E   M A I N T E N A N C E'
0022 0007 D            0531                                 P'Item number'
0023 0008 DITNBR       0544   YN Y Z                 Y
0024 0009 D            2106                                 P'Press Enter key to conX
     0010 Dtinue'
0025 0011 D            2206                                 P'Press Cmd key 7 to endX
     0012 D session'


0026 0013 C   KG                SETON                     EW
0027 0014 C   KG                GOTO END
0028 0015 C                     GET  ITEMMSTR             89    ITEM MASTER FILE
0029 0016 C   89                PUTS IM300WD4                   ADDITIONS TO MASTER
```

Figure 11-6 (Part 2 of 12). Generation Printout for Example Program 2

```
                                                                                          WSULIB   IM300W

0030 0017 C           ITNBR     COMP ' '                      69NUMBER NOT VALID
0031 0018 C   69                MSG  'INVALID ITEM NUMBERX
     0019 C'
0032 0020 C   69                PUTS IM300WD1                   REPEAT DISPLAY
0033 0021 C           ARCOD     COMP 'X'                      59DELETED ITEM NUMBER
0034 0022 C   59                MSG  'DELETED ITEM'
0035 0023 C   59                PUTS IM300WD1                   REPEAT DISPLAY
0036 0024 C           END       TAG                             END SESSION





0037 0025 SIM300WD2                            Y


0038 0026 D            0210                Y     Y          P'I T E M  M A S T E R  X
     0027 DF I L E - C H A N G E S/D E L E T I O N S'
0039 0028 D            0510                                 P'Item number'
0040 0029 DITNBR       0523Y
0041 0030 D            0710                                 P'Action'
0042 0031 DCODE        0723   YA Y                    Y
0043 0032 D            0812                                 P'Enter  C- changes'
0044 0033 D            0919                                 P'D- deletions'
0045 0034 D            2104                                 P'Press Enter key to conX
     0035 Dtinue'
0046 0036 D            2204                                 P'Press Cmd key 7 to endX
     0037 D session'
0047 0038 D            2304                                 P'Press Cmd key 8 to retX
     0039 Durn to first menu'


0048 0040 C   KG                SETON                     EW    USE CMD 7 TO END
0049 0041 C   KG                GOTO END1
0050 0042 C   KH                PUTS IM300WD1                   RETURN TO FIRST
0051 0043 C           CODE      COMP 'C'                      11CHANGES
0052 0044 C   11                PUTS IM300WD3                   GET CHANGES DISPLAY
0053 0045 C           CODE      COMP 'D'                      12DELETIONS
0054 0046 C   12                PUTS IM300WD5                   GET DELETIONS DISPLAY
0055 0047 C  N11N12             MSG  'INVALID SELECTION'        INCORRECT SELECTION
0056 0048 C           END1      TAG                             SESSION ENDED






0058 0050 SIM300WD3              Y            YY
```

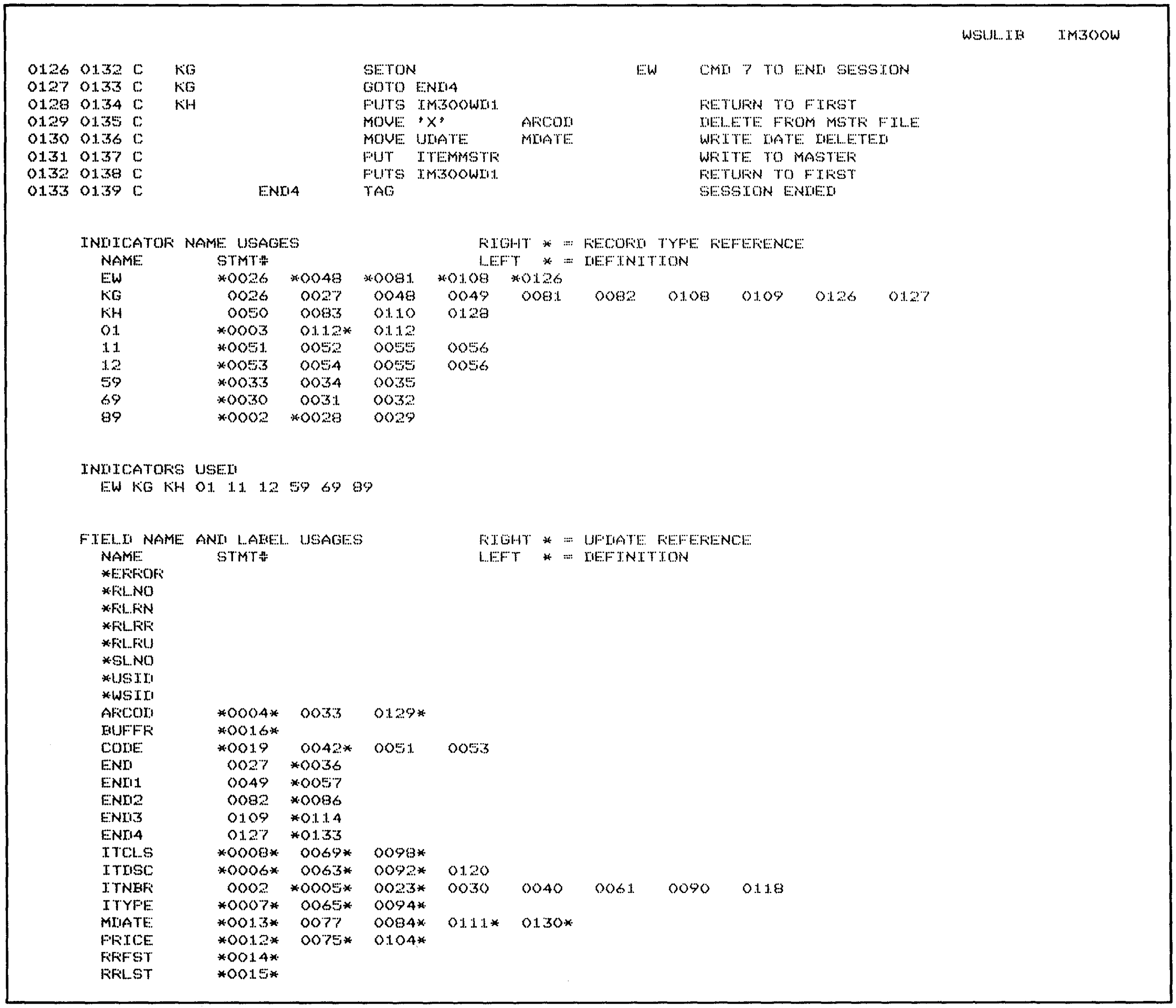
Figure 11-6 (Part 3 of 12). Generation Printout for Example Program 2

```
                                                                                          WSULIB   IM300W

0059 0051 D            0216               Y     Y          P'I T E M  M A S T E R  X
     0052 DF I L E - C H A N G E S'
0060 0053 D            0404                                P'Item number'
0061 0054 DITNBR       0423Y
0062 0055 D            0436                                P'Description'
0063 0056 DITDSC       0449Y  YB                   Y   Y
0064 0057 D            0604                                P'Item type'
0065 0058 DITYPE       0623Y  YB                     Y N
0066 0059 D            0636                                P'Unit cost'
0067 0060 DUCOST       0651Y  YN   Z                 Y
0068 0061 D            0704                                P'Item class'
0069 0062 DITCLS       0723Y  YB Y                   Y N
0070 0063 D            0736                                P'Unit weight'
0071 0064 DWEGHT       0751Y  YN   Z                 Y
0072 0065 D            0804                                P'Stock location'
0073 0066 DWHSLC       0823Y  YB Y                   Y N
0074 0067 D            0836                                P'Selling price'
0075 0068 DPRICE       0851Y  YN   Z                 Y
0076 0069 D            1204                                P'Date last maintained'
0077 0070 DMDATE       1226Y
0078 0071 D            2104                                P'Press Enter key to conX
     0072 Dtinue '
0079 0073 D            2204                                P'Press Cmd key 7 to endX
     0074 D session'
0080 0075 D            2304                                P'Press Cmd key 8 to retX
     0076 Durn to first menu'


0081 0077 C    KG                SETON                     EW    CMD 7 TO END SESSION
0082 0078 C    KG                GOTO END2
0083 0079 C    KH                PUTS IM300WD1                   RETURN TO FIRST
0084 0080 C                      MOVE UDATE     MDATE            DATE OF CHANGES
0085 0081 C                      PUT  ITEMMSTR                   CHANGES TO ITEM FILE
0086 0082 C           END2      TAG





0087 0083 SIM300WD4              Y             Y


0088 0084 D            0216               Y     Y          P'I T E M  M A S T E R  X
     0085 DF I L E - A D D I T I O N S'
0089 0086 D            0404                                P'Item number'
0090 0087 DITNBR       0423Y
0091 0088 D            0436                                P'Description'
0092 0089 DITDSC       0449   YB Y                 Y   Y
0093 0090 D            0604                                P'Item type'
0094 0091 DITYPE       0623   YB Y                   Y N
0095 0092 D            0636                                P'Unit cost'
0096 0093 DUCOST       0651   YN Y Z                 Y
```

Figure 11-6 (Part 4 of 12). Generation Printout for Example Program 2

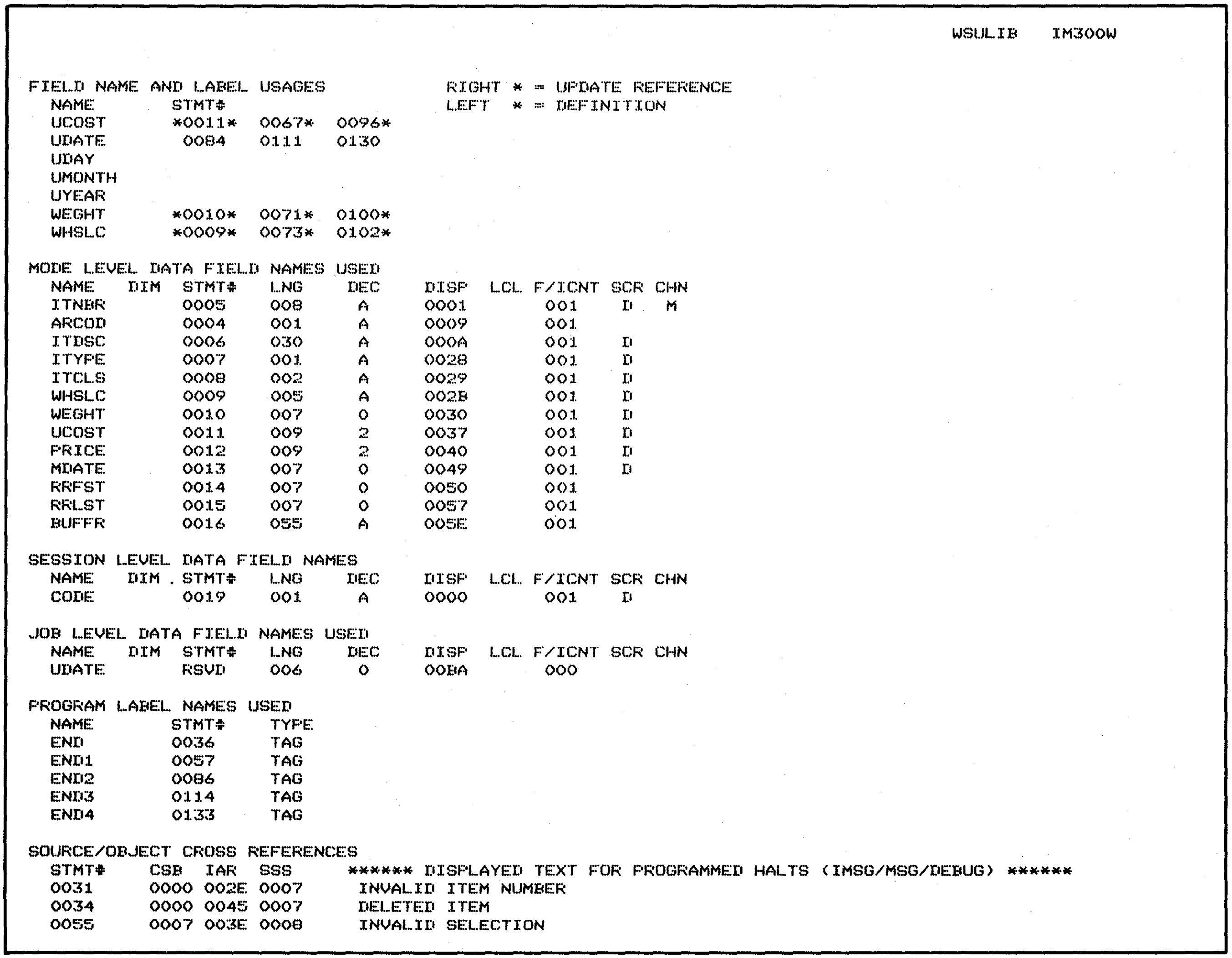
```
                                                                                        WSULIB   IM300W

0097 0094 D            0704                                P'Item class'
0098 0095 DITCLS       0723   YB Y                  Y N
0099 0096 D            0736                                P'Unit weight'
0100 0097 DWEGHT       0751   YN Y Z                Y
0101 0098 D            0804                                P'Stock location'
0102 0099 DWHSLC       0823   YB Y                  Y N
0103 0100 D            0836                                P'Selling price'
0104 0101 DPRICE       0851   YN Y Z                Y
0105 0102 D            2104                                P'Press Enter key to conX
     0103 Dtinue '
0106 0104 D            2204                                P'Press Cmd key 7 to endX
     0105 D session'
0107 0106 D            2304                                P'Press Cmd key 8 to retX
     0107 Durn to first menu'

0108 0108 C   KG                SETON                    EW    CMD 7 TO END SESSION
0109 0109 C   KG                GOTO END3
0110 0110 C   KH                PUTS IM300WD1                  RETURN TO FIRST
0111 0111 C                     MOVE UDATE     MDATE           DATE FIRST ENTERED
0112 0112 C                     PUTN ITEMMSTR            01    ITEM MASTER FILE
0113 0113 C                     PUTS IM300WD1                  RETURN TO FIRST
0114 0114 C           END3      TAG




0115 0115 SIM300WD5      Y

0116 0116 D            0218                Y     Y         P'I T E M  M A S T E R  X
     0117 DF I L E - D E L E T I O N S'
0117 0118 D            0404                                P'Item number'
0118 0119 DITNBR       0423Y
0119 0120 D            0436                                P'Description'
0120 0121 DITDSC       0449Y
0121 0122 D            0921                Y               P'THIS ENTRY IN THE MASTX
     0123 DER FILE WILL BE '
0122 0124 D            1021                Y               P'DELETED WHEN THE ENTERX
     0125 D KEY IS PRESSED'
0123 0126 D            2104                                P'Press Enter key to conX
     0127 Dtinue '
0124 0128 D            2204                                P'Press Cmd key 7 to endX
     0129 D session'
0125 0130 D            2304                                P'Press Cmd key 8 to retX
     0131 Durn to first menu'
```

Figure 11-6 (Part 5 of 12). Generation Printout for Example Program 2



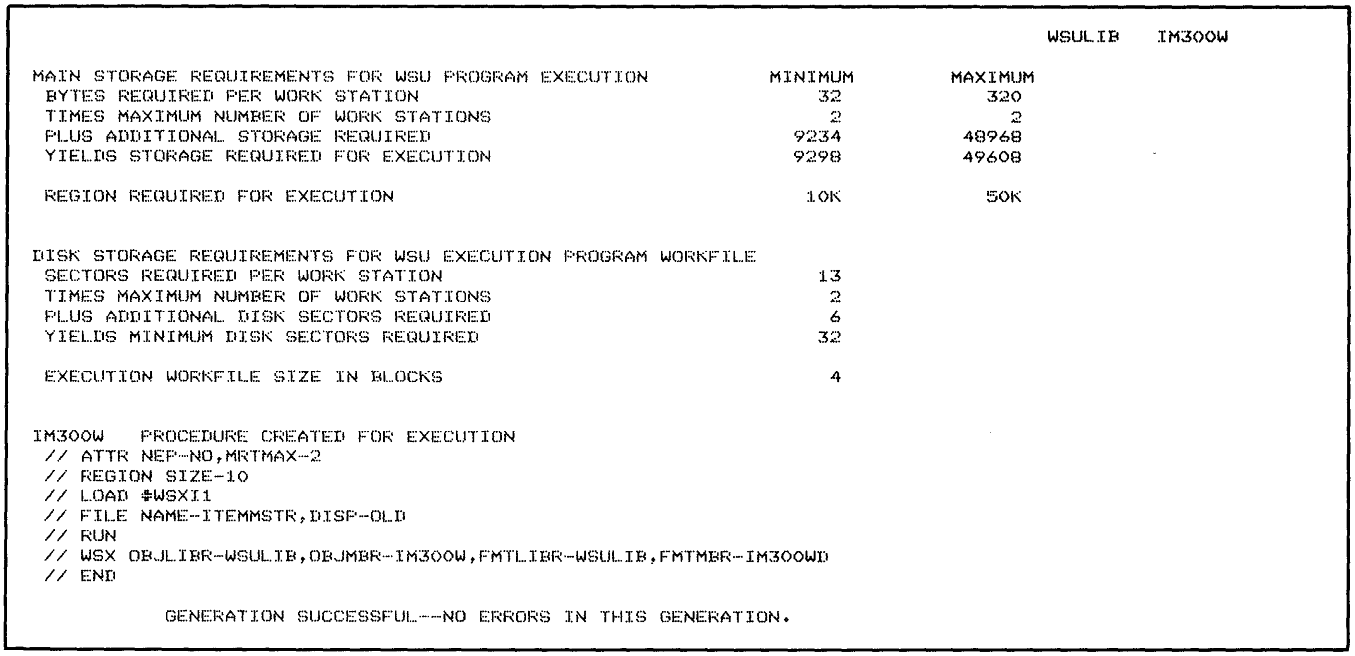
```
                                                                                         WSULIB   IM300W

0126 0132 C   KG                SETON                    EW    CMD 7 TO END SESSION
0127 0133 C   KG                GOTO END4
0128 0134 C   KH                PUTS IM300WD1                  RETURN TO FIRST
0129 0135 C                     MOVE 'X'       ARCOD           DELETE FROM MSTR FILE
0130 0136 C                     MOVE UDATE     MDATE           WRITE DATE DELETED
0131 0137 C                     PUT  ITEMMSTR                  WRITE TO MASTER
0132 0138 C                     PUTS IM300WD1                  RETURN TO FIRST
0133 0139 C           END4      TAG                            SESSION ENDED

     INDICATOR NAME USAGES                RIGHT * = RECORD TYPE REFERENCE
       NAME       STMT#                   LEFT  * = DEFINITION
       EW         *0026  *0048  *0081  *0108  *0126
       KG          0026   0027   0048   0049   0081   0082   0108   0109   0126   0127
       KH          0050   0083   0110   0128
       01         *0003   0112*  0112
       11         *0051   0052   0055   0056
       12         *0053   0054   0055   0056
       59         *0033   0034   0035
       69         *0030   0031   0032
       89         *0002  *0028   0029

     INDICATORS USED
       EW KG KH 01 11 12 59 69 89

     FIELD NAME AND LABEL USAGES          RIGHT * = UPDATE REFERENCE
       NAME       STMT#                   LEFT  * = DEFINITION
       *ERROR
       *RLNO
       *RLRN
       *RLRR
       *RLRU
       *SLNO
       *USID
       *WSID
       ARCOD      *0004*  0033   0129*
       BUFFR      *0016*
       CODE       *0019   0042*  0051   0053
       END         0027  *0036
       END1        0049  *0057
       END2        0082  *0086
       END3        0109  *0114
       END4        0127  *0133
       ITCLS      *0008*  0069*  0098*
       ITDSC      *0006*  0063*  0092*  0120
       ITNBR       0002  *0005*  0023*  0030   0040   0061   0090   0118
       ITYPE      *0007*  0065*  0094*
       MDATE      *0013*  0077   0084*  0111*  0130*
       PRICE      *0012*  0075*  0104*
       RRFST      *0014*
       RRLST      *0015*
```

Figure 11-6 (Part 6 of 12). Generation Printout for Example Program 2

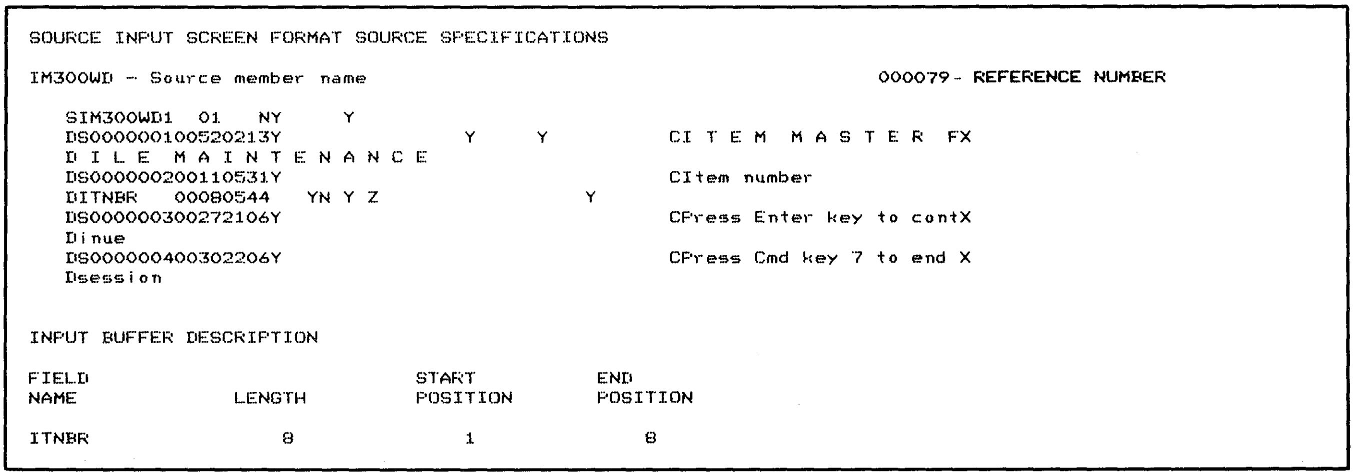


```
                                                                                    WSULIB   IM300W

FIELD NAME AND LABEL USAGES           RIGHT * = UPDATE REFERENCE
  NAME       STMT#                    LEFT  * = DEFINITION
  UCOST      *0011*  0067*  0096*
  UDATE       0084   0111   0130
  UDAY
  UMONTH
  UYEAR
  WEGHT      *0010*  0071*  0100*
  WHSLC      *0009*  0073*  0102*

MODE LEVEL DATA FIELD NAMES USED
  NAME   DIM  STMT#   LNG    DEC    DISP  LCL F/ICNT SCR CHN
  ITNBR       0005    008     A     0001       001    D   M
  ARCOD       0004    001     A     0009       001
  ITDSC       0006    030     A     000A       001    D
  ITYPE       0007    001     A     0028       001    D
  ITCLS       0008    002     A     0029       001    D
  WHSLC       0009    005     A     002B       001    D
  WEGHT       0010    007     0     0030       001    D
  UCOST       0011    009     2     0037       001    D
  PRICE       0012    009     2     0040       001    D
  MDATE       0013    007     0     0049       001    D
  RRFST       0014    007     0     0050       001
  RRLST       0015    007     0     0057       001
  BUFFR       0016    055     A     005E       001

SESSION LEVEL DATA FIELD NAMES
  NAME   DIM . STMT#  LNG    DEC    DISP  LCL F/ICNT SCR CHN
  CODE        0019    001     A     0000       001    D

JOB LEVEL DATA FIELD NAMES USED
  NAME   DIM  STMT#   LNG    DEC    DISP  LCL F/ICNT SCR CHN
  UDATE       RSVD    006     0     00BA       000

PROGRAM LABEL NAMES USED
  NAME       STMT#    TYPE
  END        0036     TAG
  END1       0057     TAG
  END2       0086     TAG
  END3       0114     TAG
  END4       0133     TAG

SOURCE/OBJECT CROSS REFERENCES
  STMT#    CSB  IAR  SSS    ****** DISPLAYED TEXT FOR PROGRAMMED HALTS (IMSG/MSG/DEBUG) ******
  0031     0000 002E 0007     INVALID ITEM NUMBER
  0034     0000 0045 0007     DELETED ITEM
  0055     0007 003E 0008     INVALID SELECTION
```

Figure 11-6 (Part 7 of 12). Generation Printout for Example Program 2

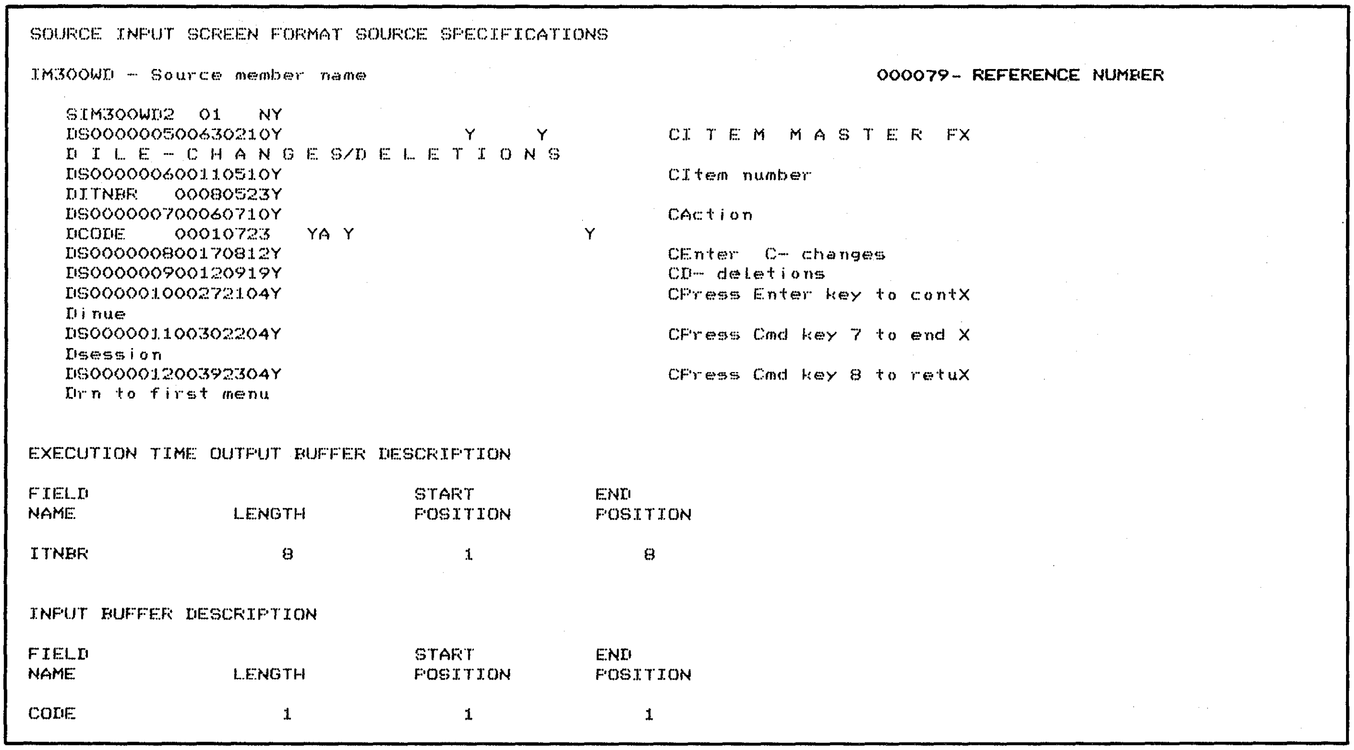


```
                                                                                    WSULIB   IM300W

MAIN STORAGE REQUIREMENTS FOR WSU PROGRAM EXECUTION          MINIMUM        MAXIMUM
 BYTES REQUIRED PER WORK STATION                                  32            320
 TIMES MAXIMUM NUMBER OF WORK STATIONS                             2              2
 PLUS ADDITIONAL STORAGE REQUIRED                               9234          48968
 YIELDS STORAGE REQUIRED FOR EXECUTION                          9298          49608

 REGION REQUIRED FOR EXECUTION                                   10K            50K


DISK STORAGE REQUIREMENTS FOR WSU EXECUTION PROGRAM WORKFILE
 SECTORS REQUIRED PER WORK STATION                                13
 TIMES MAXIMUM NUMBER OF WORK STATIONS                             2
 PLUS ADDITIONAL DISK SECTORS REQUIRED                             6
 YIELDS MINIMUM DISK SECTORS REQUIRED                             32

 EXECUTION WORKFILE SIZE IN BLOCKS                                 4


IM300W   PROCEDURE CREATED FOR EXECUTION
 // ATTR NEP-NO,MRTMAX-2
 // REGION SIZE-10
 // LOAD $WSXI1
 // FILE NAME-ITEMMSTR,DISP-OLD
 // RUN
 // WSX OBJLIBR-WSULIB,OBJMBR-IM300W,FMTLIBR-WSULIB,FMTMBR-IM300WD
 // END

           GENERATION SUCCESSFUL---NO ERRORS IN THIS GENERATION.
```



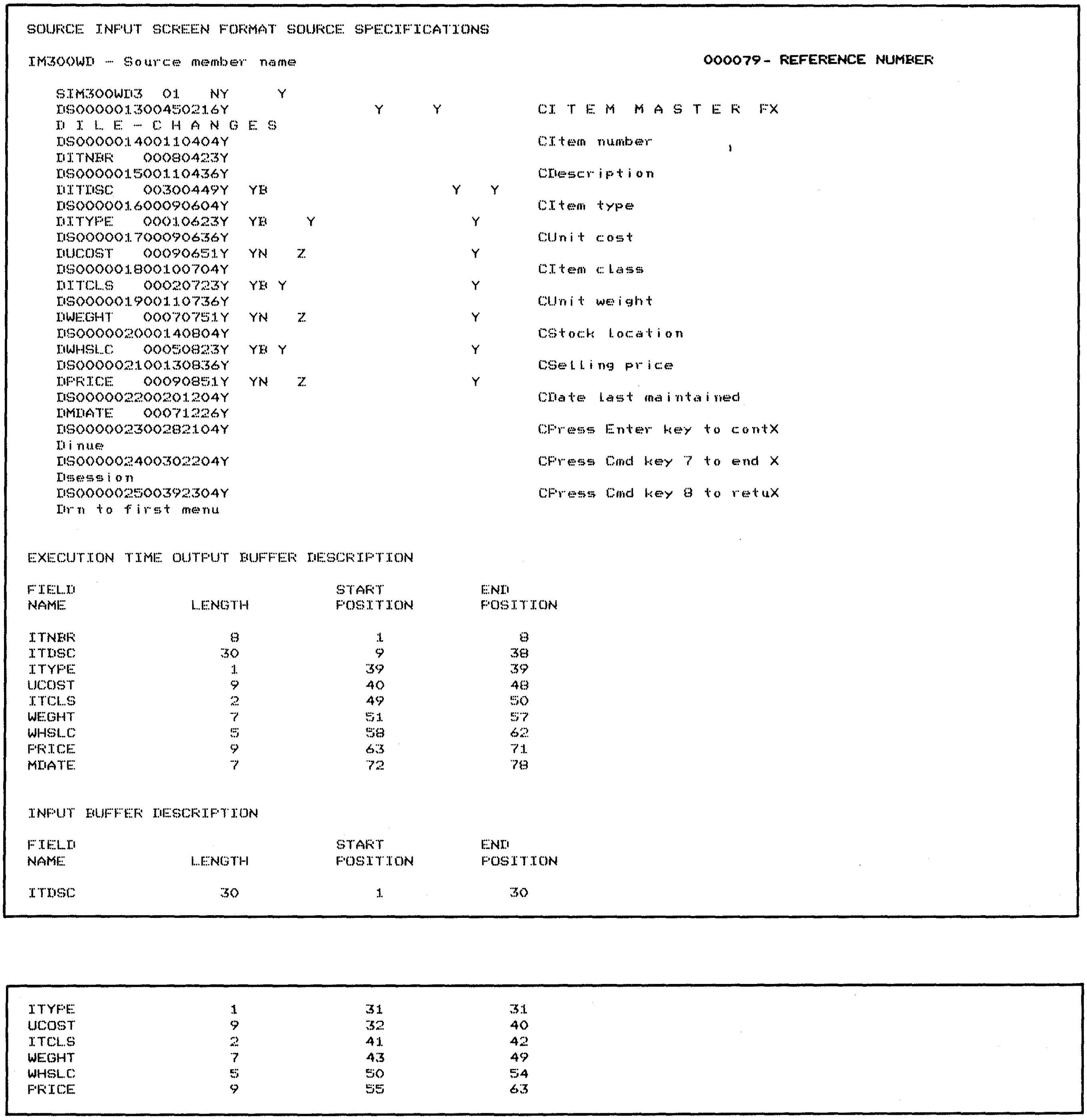
```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM300WD - Source member name                                          000079- REFERENCE NUMBER

   SIM300WD1  01   NY      Y
   DS0000000100520213Y              Y     Y          CI T E M  M A S T E R  FX
   D I L E  M A I N T E N A N C E
   DS0000000200110531Y                               CItem number
   DITNBR    00080544   YN Y Z              Y
   DS0000000300272106Y                               CPress Enter key to contX
   Dinue
   DS0000000400302206Y                               CPress Cmd key 7 to end X
   Dsession


INPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

ITNBR                8              1              8
```

Figure 11-6 (Part 8 of 12). Generation Printout for Example Program 2



```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM300WD - Source member name                                          000079- REFERENCE NUMBER

   SIM300WD2  01   NY
   DS000000500630210Y              Y     Y          CI T E M   M A S T E R  FX
   D I L E - C H A N G E S/D E L E T I O N S
   DS000000600110510Y                               CItem number
   DITNBR   00080523Y
   DS000000700060710Y                               CAction
   DCODE     00010723   YA Y              Y
   DS000000800170812Y                               CEnter  C- changes
   DS000000900120919Y                               CD- deletions
   DS000001000272104Y                               CPress Enter key to contX
   Dinue
   DS000001100302204Y                               CPress Cmd key 7 to end X
   Dsession
   DS000001200392304Y                               CPress Cmd key 8 to retuX
   Drn to first menu

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                            START         END
NAME            LENGTH           POSITION      POSITION

ITNBR                8               1             8

INPUT BUFFER DESCRIPTION

FIELD                            START         END
NAME            LENGTH           POSITION      POSITION

CODE                 1               1             1
```

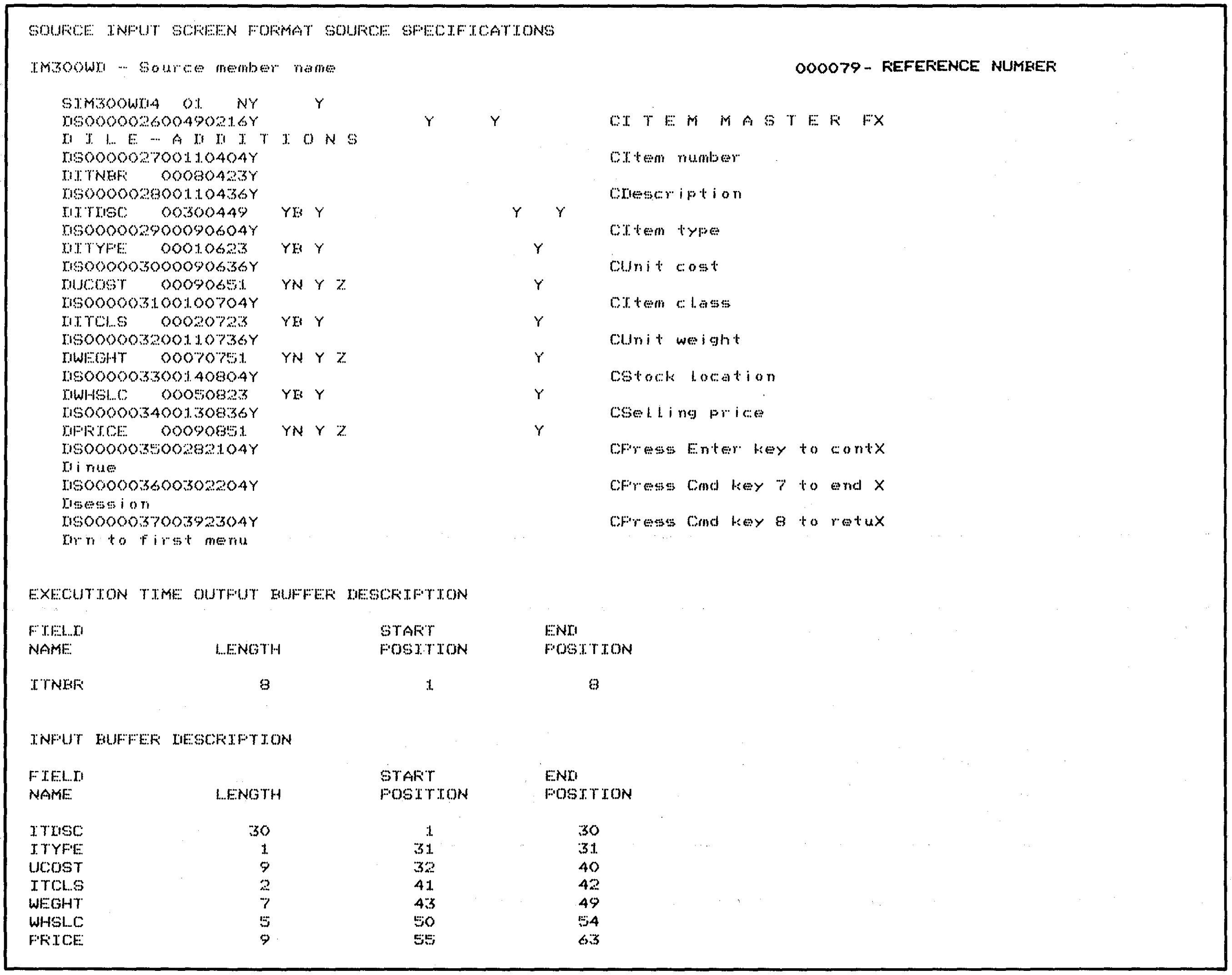
Figure 11-6 (Part 9 of 12). Generation Printout for Example Program 2



```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM300WD - Source member name                                           000079- REFERENCE NUMBER

   SIM300WD3  01   NY      Y
   DS000001300450216Y                 Y     Y         CI T E M  M A S T E R  FX
   D I L E - C H A N G E S
   DS000001400110404Y                                 CItem number
   DITNBR    00080423Y
   DS000001500110436Y                                 CDescription
   DITDSC    00300449Y  YB                  Y   Y
   DS000001600090604Y                                 CItem type
   DITYPE    00010623Y  YB    Y                 Y
   DS000001700090636Y                                 CUnit cost
   DUCOST    00090651Y  YN   Z                  Y
   DS000001800100704Y                                 CItem class
   DITCLS    00020723Y  YB Y                    Y
   DS000001900110736Y                                 CUnit weight
   DWEGHT    00070751Y  YN   Z                  Y
   DS000002000140804Y                                 CStock location
   DWHSLC    00050823Y  YB Y                    Y
   DS000002100130836Y                                 CSelling price
   DPRICE    00090851Y  YN   Z                  Y
   DS000002200201204Y                                 CDate last maintained
   DMDATE    00071226Y
   DS000002300282104Y                                 CPress Enter key to contX
   Dinue
   DS000002400302204Y                                 CPress Cmd key 7 to end X
   Dsession
   DS000002500392304Y                                 CPress Cmd key 8 to retuX
   Drn to first menu

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

ITNBR                 8              1              8
ITDSC                30              9             38
ITYPE                 1             39             39
UCOST                 9             40             48
ITCLS                 2             49             50
WEGHT                 7             51             57
WHSLC                 5             58             62
PRICE                 9             63             71
MDATE                 7             72             78

INPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

ITDSC                30              1             30
```

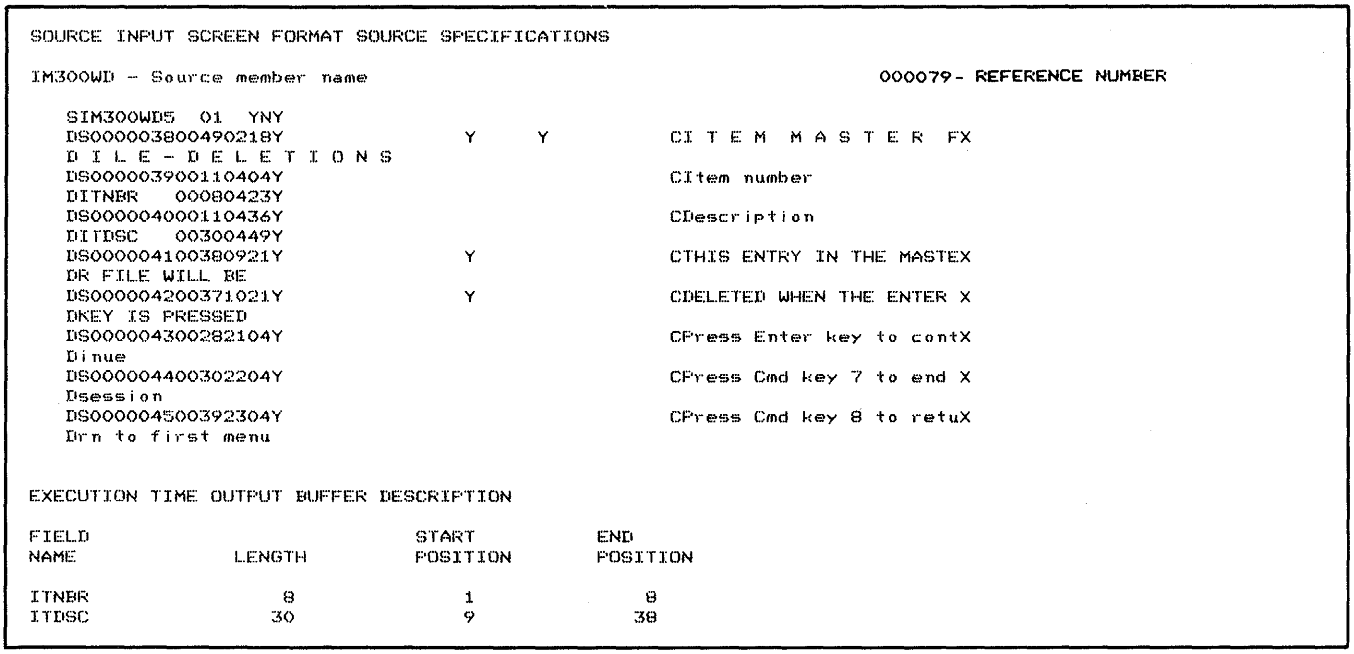
```
ITYPE                 1             31             31
UCOST                 9             32             40
ITCLS                 2             41             42
WEGHT                 7             43             49
WHSLC                 5             50             54
PRICE                 9             55             63
```

Figure 11-6 (Part 10 of 12). Generation Printout for Example Program 2

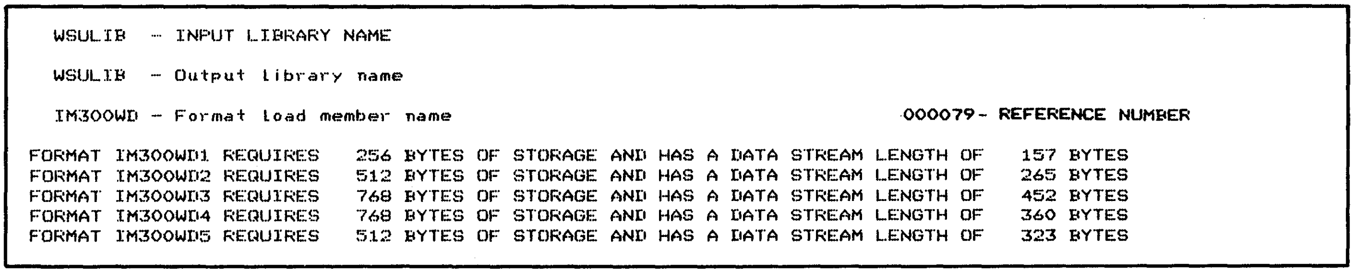


```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM300WD - Source member name                                          000079- REFERENCE NUMBER

   SIM300WD4  01   NY      Y
   DS0000026004902l6Y                    Y     Y          CI T E M  M A S T E R  FX
   D I L E - A D D I T I O N S
   DS0000027001l0404Y                                     CItem number
   DITNBR   00080423Y
   DS000002800110436Y                                     CDescription
   DITDSC   00300449   YB Y                 Y   Y
   DS000002900090604Y                                     CItem type
   DITYPE   00010623   YB Y               Y
   DS000003000090636Y                                     CUnit cost
   DUCOST   00090651   YN Y Z             Y
   DS000003100100704Y                                     CItem class
   DITCLS   00020723   YB Y               Y
   DS000003200110736Y                                     CUnit weight
   DWEGHT   00070751   YN Y Z             Y
   DS000003300140804Y                                     CStock location
   DWHSLC   00050823   YB Y               Y
   DS000003400130836Y                                     CSelling price
   DPRICE   00090851   YN Y Z             Y
   DS000003500282104Y                                     CPress Enter key to contX
   Dinue
   DS000003600302204Y                                     CPress Cmd key 7 to end X
   Dsession
   DS000003700392304Y                                     CPress Cmd key 8 to retuX
   Drn to first menu

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

ITNBR                 8              1              8

INPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

ITDSC                30              1             30
ITYPE                 1             31             31
UCOST                 9             32             40
ITCLS                 2             41             42
WEGHT                 7             43             49
WHSLC                 5             50             54
PRICE                 9             55             63
```

Figure 11-6 (Part 11 of 12). Generation Printout for Example Program 2



```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

IM300WD - Source member name                                          000079- REFERENCE NUMBER

   SIM300WD5  01  YNY
   DS00000380049021 8Y                 Y     Y          CI T E M   M A S T E R  FX
   D I L E - D E L E T I O N S
   DS0000039001104 04Y                                  CItem number
   DITNBR    00080423Y
   DS0000040001104 36Y                                  CDescription
   DITDSC    00300449Y
   DS0000041003809 21Y                 Y                CTHIS ENTRY IN THE MASTEX
   DR FILE WILL BE
   DS0000042003710 21Y                 Y                CDELETED WHEN THE ENTER X
   DKEY IS PRESSED
   DS0000043002821 04Y                                  CPress Enter key to contX
   Dinue
   DS0000044003022 04Y                                  CPress Cmd key 7 to end X
   Dsession
   DS0000045003923 04Y                                  CPress Cmd key 8 to retuX
   Drn to first menu


EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

ITNBR               8              1              8
ITDSC              30              9             38
```



```
  WSULIB   - INPUT LIBRARY NAME

  WSULIB   - Output library name

  IM300WD  - Format load member name                                    000079- REFERENCE NUMBER

FORMAT IM300WD1 REQUIRES   256 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   157 BYTES
FORMAT IM300WD2 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   265 BYTES
FORMAT IM300WD3 REQUIRES   768 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   452 BYTES
FORMAT IM300WD4 REQUIRES   768 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   360 BYTES
FORMAT IM300WD5 REQUIRES   512 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   323 BYTES
```

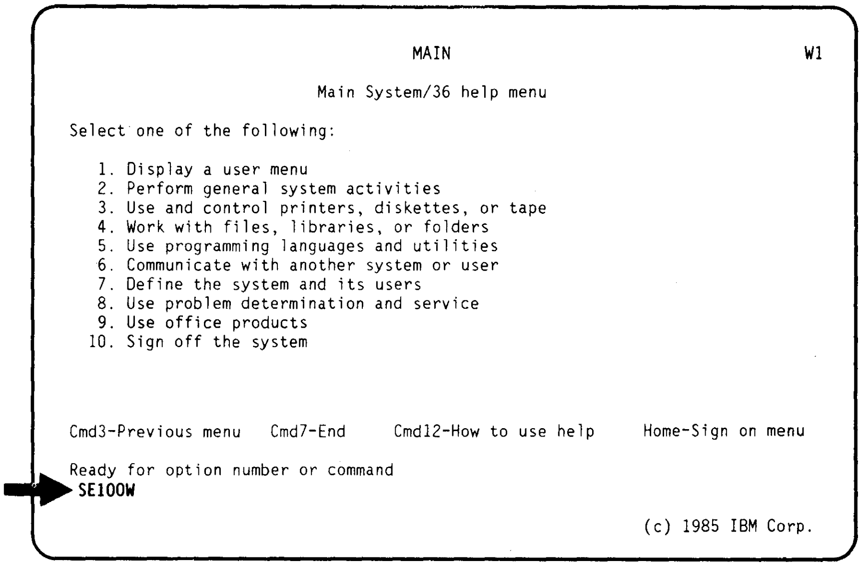
Figure 11-6 (Part 12 of 12). Generation Printout for Example Program 2

Example Program 3: Sales Entry

This program, named SE100W, allows one operator to enter monthly sales amounts. Input from the operator consists of a date and a sales amount. WSU uses array processing to display a sales summary listing the months, sales totals, and a yearly total.

Running SE100W

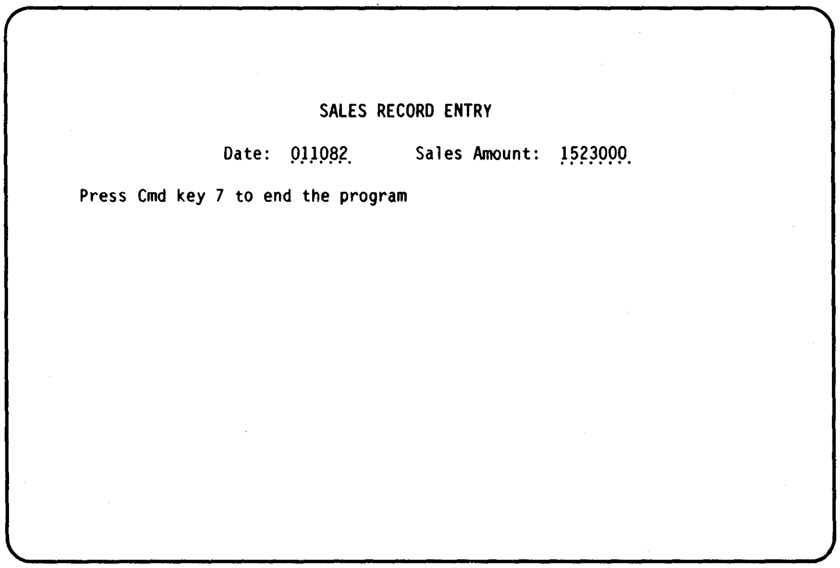
To initiate WSU program SE100W, the operator enters the procedure name, SE100W.



```
                              MAIN                                 W1

                     Main System/36 help menu

Select one of the following:

   1. Display a user menu
   2. Perform general system activities
   3. Use and control printers, diskettes, or tape
   4. Work with files, libraries, or folders
   5. Use programming languages and utilities
   6. Communicate with another system or user
   7. Define the system and its users
   8. Use problem determination and service
   9. Use office products
  10. Sign off the system



Cmd3-Previous menu   Cmd7-End     Cmd12-How to use help     Home-Sign on menu

Ready for option number or command
SE100W
                                                          (c) 1985 IBM Corp.
```

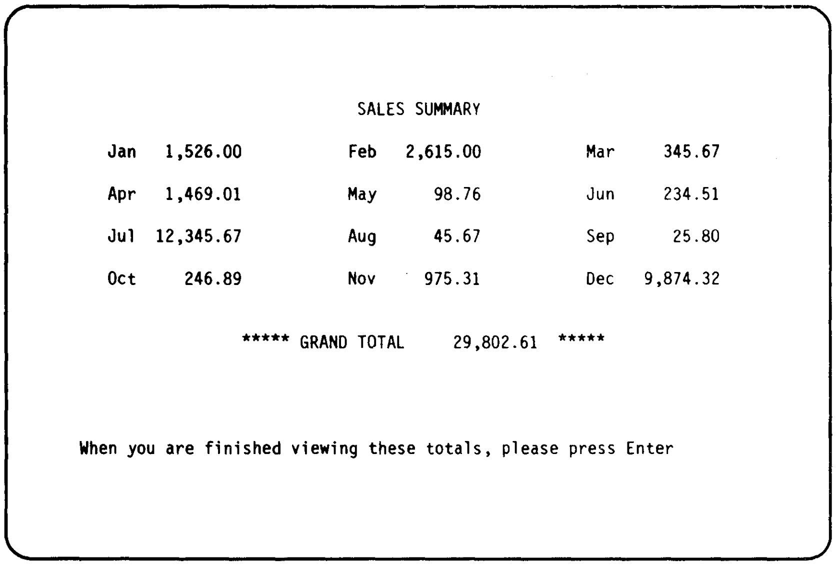
This display occurs when an operator initiates the WSU program SE100W.

The operator enters the sales date (MMDDYY) and the sales amount. The information is put into a transaction file named SALESFLE.



```
                        SALES RECORD ENTRY

             Date:  011082       Sales Amount:  1523000

   Press Cmd key 7 to end the program
```

This display (SALES SUMMARY) appears when the operator presses command key 7 to end the job after the last amount is entered. No entries can be made to this summary. By pressing the Enter key after viewing the summary, the operator can end this program. If you would like a copy of this summary, you can use the Print key to print the display.



```
                              SALES SUMMARY

    Jan   1,526.00            Feb  2,615.00            Mar     345.67

    Apr   1,469.01            May     98.76            Jun     234.51

    Jul  12,345.67            Aug     45.67            Sep      25.80

    Oct     246.89            Nov    975.31            Dec   9,874.32

                   ***** GRAND TOTAL     29,802.61  *****



 When you are finished viewing these totals, please press Enter
```

Coding Forms

In this example, the file definition is in two source members:

- FILEDEF contains F- and I-specifications for the transaction file, SALESFLE.
- FILEDEF contains F- and I-specifications for the master file, MONTHS.

Figure 11-7 shows the file definition for example program 3.

File Description Specifications

For the valid entries for a system, refer to the RPG reference manual for that system.

Line	Form Type	Filename	File Type (I/O/U/C/D)	Record Length
0 2	F	SALESFLE	I	98

IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Program

Programmer

Date

Keying Instruction: Graphic, Key

Card Electro Number

Page of

Program Identification

Line	Form Type	Filename or Record Name	Record Identifying Indicator, or DS	Record Identification Codes 1: Position	C/Z/D	Character	Field Location From	Field Location To	Decimal Positions	RPG Field Name
0 1	I	SALESFLE	01	1	C	S				
0 2	I						1	1		RCODE
0 3	I						2	7	0	DATE
0 4	I						8	14	2	SALAMT
0 5	I		02	1	C	T				
0 6	I						1	1		TCODE
0 7	I						2	85	2	SALTOT

File Description Specifications

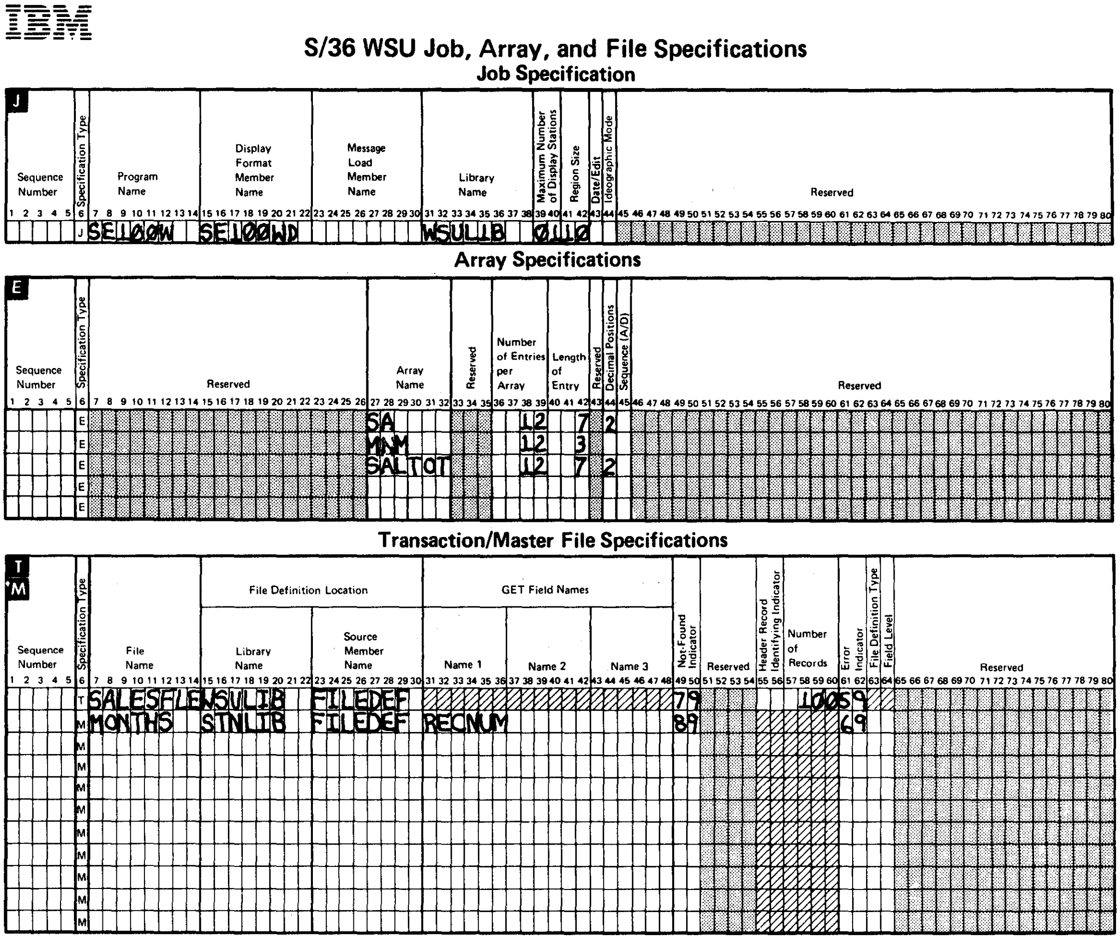
For the valid entries for a system, refer to the RPG reference manual for that system.

Line	Form Type	Filename	File Type (I/O/U/C/D)	Record Length
0 2	F	MONTHS	I	40

Line	Form Type	Filename or Record Name	Record Identifying Indicator, or DS	Record Identification Codes 1: Position	C/Z/D	Character	Field Location From	Field Location To	Decimal Positions	RPG Field Name
0 1	I	MONTHS	01	37	C	M				
0 2	I						1	36		MNM
0 3	I						37	37		CODE
0 4	I						38	40	0	RECNUM

Figure 11-7. File Definition for Example Program 3

The source program is in a source member named SE100W. Figure 11-8 shows the WSU specifications for SE100W.



IBM

S/36 WSU Job, Array, and File Specifications

Job Specification

Sequence Number 1–5	Specification Type 6	Program Name 7–14	Display Format Member Name 15–22	Message Load Member Name 23–30	Library Name 31–38	Maximum Number of Display Stations 39–40	Region Size 41–42	Date/Edit 43	Ideographic Mode 44	Reserved 45–80
	J	SE100W	SE100WD		WSULIB	01	10			

Array Specifications

Sequence Number 1–5	Specification Type 6	Reserved 7–26	Array Name 27–32	Reserved 33–35	Number of Entries per Array 36–39	Length of Entry 40–42	Reserved 43	Decimal Positions 44	Sequence (A/D) 45	Reserved 46–80
	E		SA		12	7		2		
	E		MNM		12	3				
	E		SALTOT		12	7		2		
	E									
	E									

Transaction/Master File Specifications

Sequence Number 1–5	Specification Type 6	File Name 7–14	File Definition Location: Library Name 15–22	File Definition Location: Source Member Name 23–30	GET Field Names: Name 1 31–36	GET Field Names: Name 2 37–42	GET Field Names: Name 3 43–48	Not-Found Indicator 49–50	Reserved 51–54	Header Record Identifying Indicator 55–56	Number of Records 57–60	Error Indicator 61–62	File Definition Type 63	Field Level 64	Reserved 65–80
	T	SALESFLE	WSULIB	FILEDEF				79			100	59			
	M	MONTHS	STNLIB	FILEDEF	RECNUM			89				69			
	M														
	M														
	M														
	M														
	M														
	M														
	M														
	M														
	M														

Figure 11-8 (Part 1 of 3). WSU Specifications for Example Program 3

S/36 WSU Processing Specifications

C Sequence Number (1–5)	Specification Type (6)	Processing Function (7–8)	Conditioning Indicators: Not / And Not / And Not (9–17)	Factor 1 (18–27)	Operation (28–32)	Factor 2 (33–42)	Result Field / Literal: Name (43–48)	Length (49–51)	Decimal Pos (52)	Half Adjust (H) (53)	Resulting Indicators: Arithmetic Plus Minus Zero / Compare High 1>2 Low 1<2 Equal 1=2 (54–59)	Comments (60–80)
	C	IJ			Z-ADD	1	RECNUN					
	C	IJ			GET	MONTHS						GET MONTHS MASTER FLE
	C	IJ			GETPR	SALESFLE						GET AMNT TRANS FILE
	C	IJ			MOVE	SALTOT	SA					MOVE AMNTS TO ARRAY

System/36 Display Format Specifications

Display Control Specification

S Sequence Number (1–5)	Specification Type (6)	Display Format Name (7–14)	Format ID (WSU Only) (15–16)	Start Line Number (17–18)	Number of Lines to Clear (19–20)	Lowercase (21)	Return Input (22)	Reset Keyboard (WSU Only) (23–24)	Sound Alarm (25–26)	Enable Function Keys (27)	Enable Command Keys (28)	Blink Cursor (29–30)	Erase Input Fields (31–32)	Override Fields (33–34)	Suppress Input (35–36)	Null Fill (37–38)	Reserved (39)	Right-to-Left Display (40)	WSU Only: Enter Mode Sequence Start (41)	End (42)	Entry Required (43)	Repeat (44)	Reserved (45)	Priority (46)	Preprocess (47)	Review Mode Record Identifying Indicators 1 2 3 (48–53)	Insert Mode Record Identifying Indicators 1 2 3 (54–59)	Reserved (60–63)	Key Mask (64–79)	Reserved (80)
	S	SEL00WD1		01	24														Y	Y										

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

Field Definition Specification

D Sequence Number (1–5)	Specification Type (6)	WSU Field Name / Field Name (7–14)	WSU Arrays: Elements per Row or Column, Fill Direction, Line Skip Factor, Horizontal Positions Space Factor / Field Length (15–18)	Starting Location: Line Number (19–20)	Horizontal Position (21–22)	Output Data (23–24)	Edit Code (WSU Only) (25)	Input Allowed (26)	Data Type (27)	Mandatory Fill (28)	Mandatory Entry (29)	Self-Check (30)	Adjust/Fill (31)	Position Cursor (32–33)	Enable Dup (34)	Controlled Field Exit (35)	Auto Record Advance (36)	Protect Field (37–38)	High Intensity (39–40)	Blink Field (41–42)	Nondisplay (43–44)	Reverse Image (45–46)	Underline (47–48)	Column Separators (49)	Reserved (50)	Lowercase (51)	Reserved (52–55)	Constant Type (56)	Constant Data (57–79)	Continuation (80)
	D			04	31														Y									P	‘SALES RECORD ENTRY’	
	D			06	21														Y									P	‘Date:’	
	D	DATE		06	27			Y	N				Z											Y						
	D			06	41														Y									P	‘Sales Amount:’	
	D	SALAMT		06	55			Y	N				Z				Y							Y						
	D			08	06																	Y						P	‘Press Cmd key 7 to end-	
	D	the program’																												

Figure 11-8 (Part 2 of 3). WSU Specifications for Example Program 3

S/36 WSU Processing Specifications

Sequence Number	Specification Type	Processing Function	Conditioning Indicators	And	And	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Pos	Half Adjust (H)	Plus / High 1>2	Minus / Low 1<2	Zero / Equal 1=2	Comments
	C		KG				SETON						EW			OPERATOR SIGNALED
	C		KG				GOTO	END								END-OF-JOB
	C						MOVEL	DATE	MTH	2	0					EST. THE SALES MONTH
	C					MTH	RANGE	1	12						50	TEST FOR VALID MONTH.
	C		N50				MSG	'INVALID DATE'								MTH NOT JAN THRU DEC
	C					SALAMT	ADD	SA,MTH	SA,MTH							ACCUMULATE THE SALE
	C						MOVE	'S'	RCODE							EST. THE RECORD CODE
	C						PUT	SALESFLE					01			WRITE TRANS. RECORD
	C					END	TAG									
	C		KG				XFOOT	SA	TOTAL	10	2					TOTAL OF ALL SALES
	C		KG				MOVE	'T'	TCODE							
	C		KG				Z-ADD	SA	SALTOT							SAVE MONTHLY TOTALS
	C		KG				PUT	SALESFLE					02			MOVE TO TRANS FILE

System/36 Display Format Specifications

Display Control Specification

Sequence Number	Specification Type	Display Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Null Fill	Reserved	Right-to-Left Display	Enter Mode Sequence Start	Enter Mode Sequence End	Entry Required	Repeat	Reserved	Priority	Preprocess	Review Mode Record Identifying Indicators 1	2	3	Insert Mode Record Identifying Indicators 1	2	3	Reserved	Key Mask	Reserved
	S	SE100WD2	EW	01	24																													

The H specifications are optional. If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format. Use the other side of this form to code the H specifications.

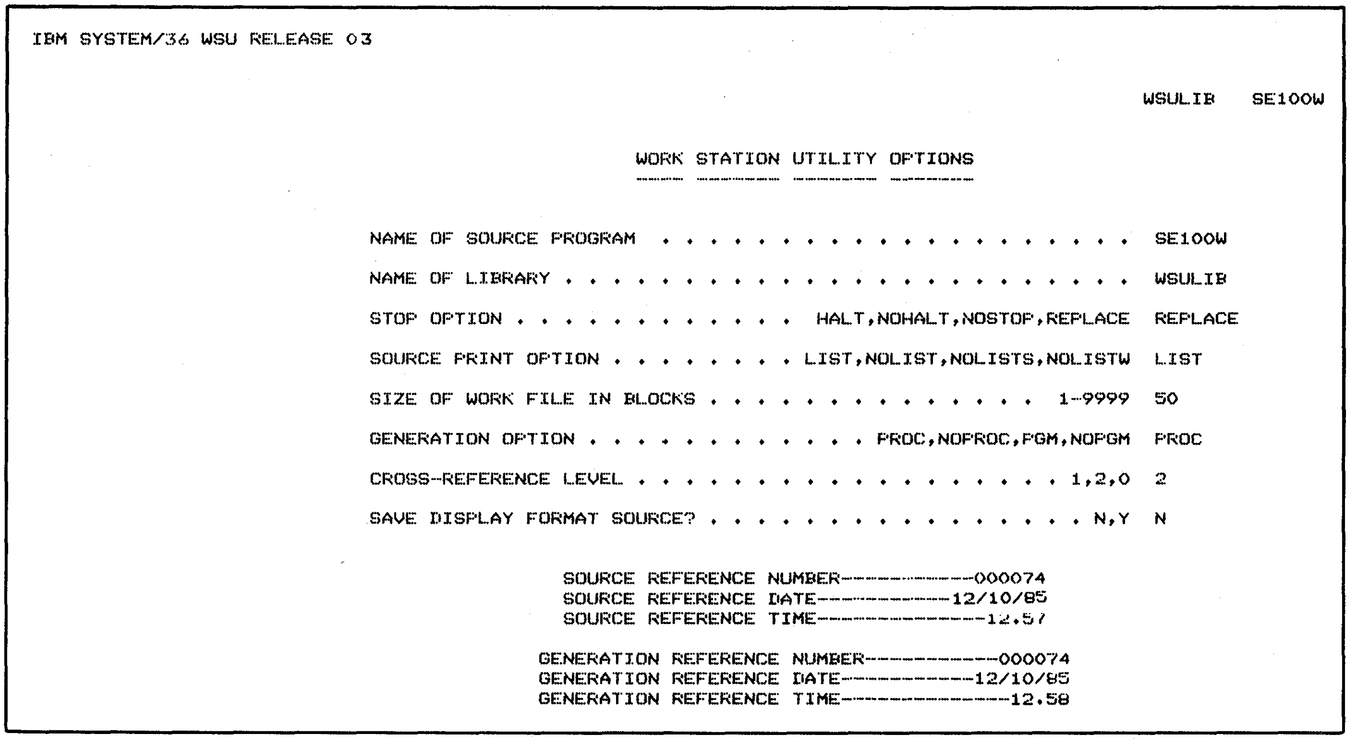
Field Definition Specification

Sequence Number	Specification Type	WSU Field Name / Field Name	Elements per Row or Column	Fill Direction	Line Skip Factor	Horizontal Positions Space Factor	Field Length	Line Number	Horizontal Position	Output Data	Edit Code (WSU Only)	Input Allowed	Data Type	Mandatory Fill	Mandatory Entry	Self-Check	Adjust/Fill	Position Cursor	Enable Dup	Controlled Field Exit	Auto Record Advance	Protect Field	High Intensity	Blink Field	Nondisplay	Reverse Image	Underline	Column Separators	Reserved	Lowercase	Reserved	Constant Type	Constant Data	Continuation
	D							04	35														Y									P	'SALES SUMMARY'	
	D	SA	03	R	1	15		06	14	Y	J																							
	D	MNM	03	R	1	22		06	09	Y																								
	D							15	23														Y									P	'XXXXX GRAND TOTAL'	
	D	TOTAL						15	41	Y	J																							
	D							15	56														Y									P	'XXXXX'	
	D							20	06																	Y						P	'When you are finished	-
	D	viewing these totals, please press Enter'																																

Figure 11-8 (Part 3 of 3). WSU Specifications for Example Program 3

Program Generation Printout

Figure 11-9 shows the generation printout for SE100W.

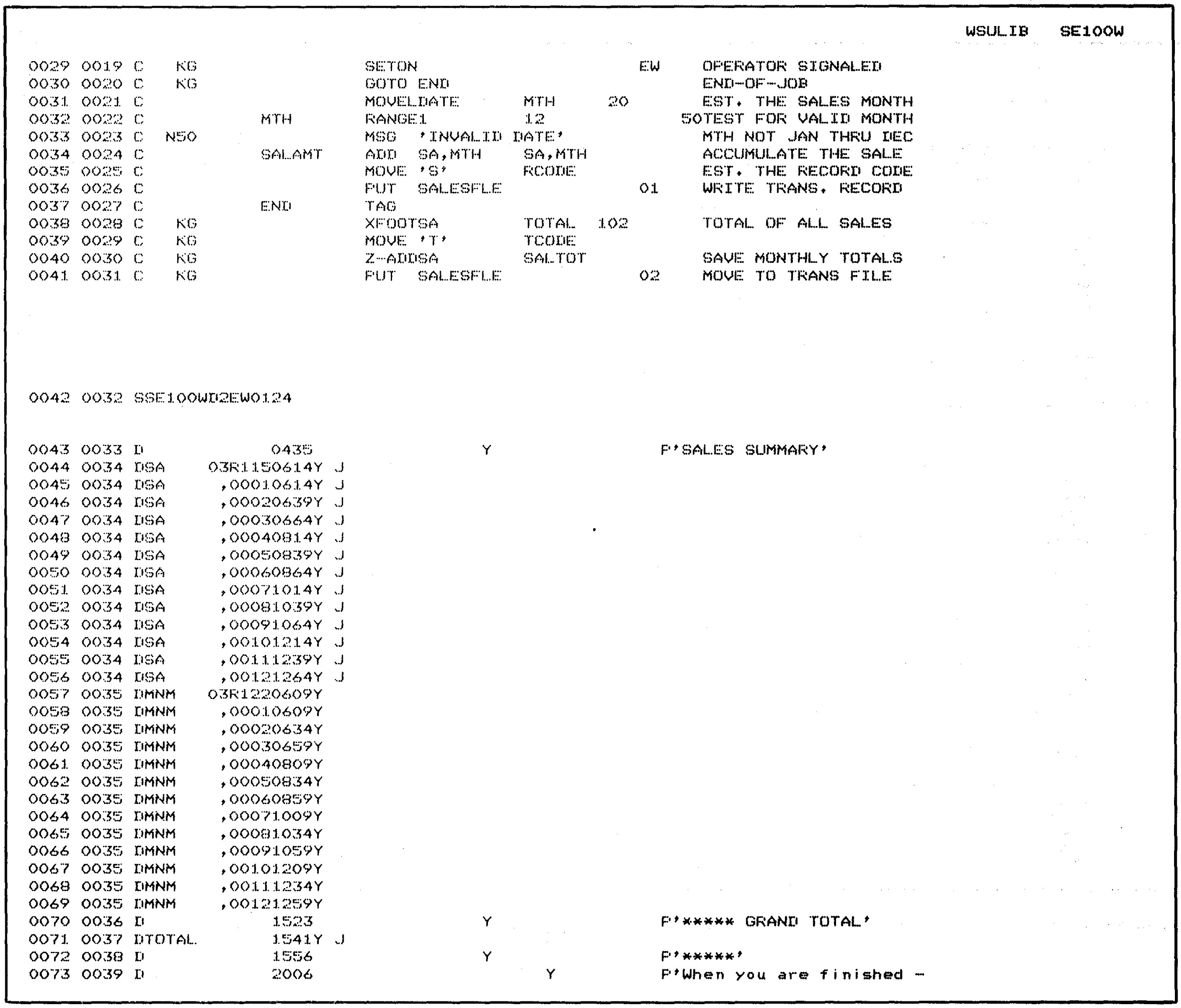


```
IBM SYSTEM/36 WSU RELEASE 03

                                                                                                   WSULIB   SE100W

                                                  WORK STATION UTILITY OPTIONS
                                                  ---- ------- ------- -------

                              NAME OF SOURCE PROGRAM  . . . . . . . . . . . . . . . . . . . .  SE100W

                              NAME OF LIBRARY . . . . . . . . . . . . . . . . . . . . . . . .  WSULIB

                              STOP OPTION . . . . . . . . . . . .  HALT,NOHALT,NOSTOP,REPLACE  REPLACE

                              SOURCE PRINT OPTION . . . . . . . . LIST,NOLIST,NOLISTS,NOLISTW  LIST

                              SIZE OF WORK FILE IN BLOCKS . . . . . . . . . . . . . .  1-9999  50

                              GENERATION OPTION . . . . . . . . . . . . PROC,NOPROC,PGM,NOPGM  PROC

                              CROSS-REFERENCE LEVEL . . . . . . . . . . . . . . . . . . 1,2,0  2

                              SAVE DISPLAY FORMAT SOURCE? . . . . . . . . . . . . . . . . N,Y  N

                                              SOURCE REFERENCE NUMBER------------000074
                                              SOURCE REFERENCE DATE------------12/10/85
                                              SOURCE REFERENCE TIME---------------12.57

                                            GENERATION REFERENCE NUMBER-----------000074
                                            GENERATION REFERENCE DATE-----------12/10/85
                                            GENERATION REFERENCE TIME--------------12.58
```

Figure 11-9 (Part 1 of 8). Generation Printout for Example Program 3

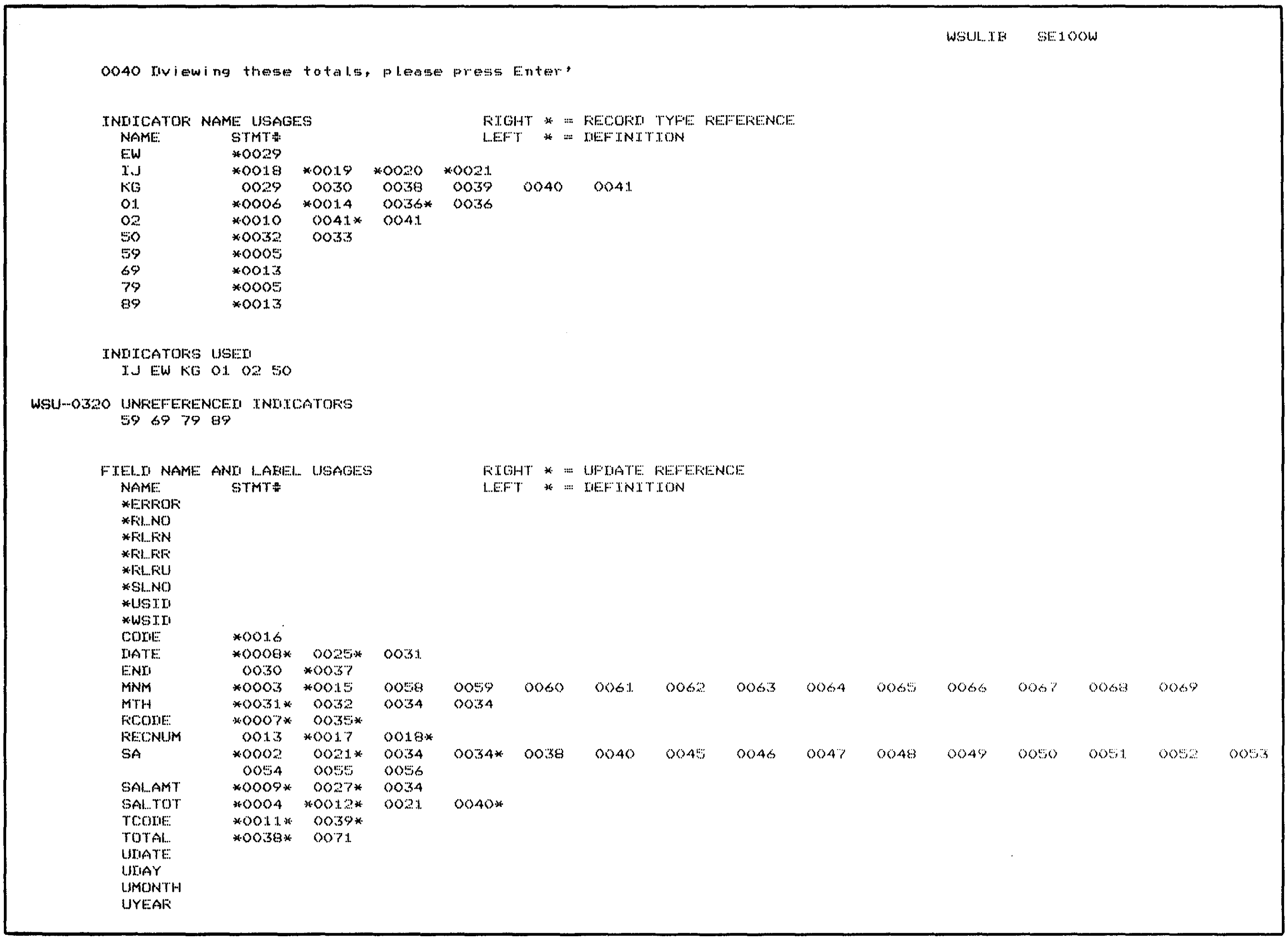
```
                                                                                          WSULIB   SE100W


0001 0001 JSE100W  SE100WD         WSULIB  0110
0002 0002 E                    SA         12  7 2             SALES ARRAY
0003 0003 E                    MNM        12  3
0004 0004 E                    SALTOT     12  7 2


0005 0005 TSALESFLEWSULIB  FILEDEF                   79       10059
             0025 FSALESFLEI          98


0006         0026 ISALESFLE    01   1 CS
0007         0027 I                                        1   1 RCODE
0008         0028 I                                        2   70DATE
0009         0029 I                                        8  142SALAMT
0010         0030 I            02   1 CT
0011         0031 I                                        1   1 TCODE
0012         0032 I                                        2  852SALTOT


0013 0006 MMONTHS  STNLIB  FILEDEF RECNUM            89          69
             0033 FMONTHS  I          40


0014         0034 IMONTHS      01  37 CM
0015         0035 I                                        1  36 MNM
0016         0036 I                                       37  37 CODE
0017         0037 I                                       38  400RECNUM


0018 0007 CIJ                   Z-ADD1         RECNUM
0019 0008 CIJ                   GET  MONTHS                     GET MONTHS MASTER FLE
0020 0009 CIJ                   GETPRSALESFLE                   GET AMNT TRANS FILE
0021 0010 CIJ                   MOVE SALTOT    SA               MOVE AMNTS TO ARRAY




0022 0011 SSE100WD1  0124                    YY


0023 0012 D            0431                Y                P'SALES RECORD ENTRY'
0024 0013 D            0621                Y                P'Date:'
0025 0014 DDATE        0627   YN   Z                 Y
0026 0015 D            0641                Y                P'Sales Amount:'
0027 0016 DSALAMT      0655   YN   Z    Y            Y
0028 0017 D            0806                      Y          P'Press Cmd key 7 to end-
     0018 D the program'
```

Figure 11-9 (Part 2 of 8). Generation Printout for Example Program 3



```
                                                                                        WSULIB   SE100W
0029 0019 C   KG                SETON                     EW    OPERATOR SIGNALED
0030 0020 C   KG                GOTO END                        END-OF-JOB
0031 0021 C                     MOVELDATE      MTH    20        EST. THE SALES MONTH
0032 0022 C           MTH       RANGE1         12             50TEST FOR VALID MONTH
0033 0023 C  N50                MSG  'INVALID DATE'             MTH NOT JAN THRU DEC
0034 0024 C           SALAMT    ADD  SA,MTH    SA,MTH           ACCUMULATE THE SALE
0035 0025 C                     MOVE 'S'       RCODE            EST. THE RECORD CODE
0036 0026 C                     PUT  SALESFLE             01    WRITE TRANS. RECORD
0037 0027 C           END       TAG
0038 0028 C   KG                XFOOTSA        TOTAL 102        TOTAL OF ALL SALES
0039 0029 C   KG                MOVE 'T'       TCODE
0040 0030 C   KG                Z-ADDSA        SALTOT           SAVE MONTHLY TOTALS
0041 0031 C   KG                PUT  SALESFLE             02    MOVE TO TRANS FILE







0042 0032 SSE100WD2EW0124


0043 0033 D            0435                Y                P'SALES SUMMARY'
0044 0034 DSA    03R1150614Y J
0045 0034 DSA     ,00010614Y J
0046 0034 DSA     ,00020639Y J
0047 0034 DSA     ,00030664Y J
0048 0034 DSA     ,00040814Y J
0049 0034 DSA     ,00050839Y J
0050 0034 DSA     ,00060864Y J
0051 0034 DSA     ,00071014Y J
0052 0034 DSA     ,00081039Y J
0053 0034 DSA     ,00091064Y J
0054 0034 DSA     ,00101214Y J
0055 0034 DSA     ,00111239Y J
0056 0034 DSA     ,00121264Y J
0057 0035 DMNM   03R1220609Y
0058 0035 DMNM    ,00010609Y
0059 0035 DMNM    ,00020634Y
0060 0035 DMNM    ,00030659Y
0061 0035 DMNM    ,00040809Y
0062 0035 DMNM    ,00050834Y
0063 0035 DMNM    ,00060859Y
0064 0035 DMNM    ,00071009Y
0065 0035 DMNM    ,00081034Y
0066 0035 DMNM    ,00091059Y
0067 0035 DMNM    ,00101209Y
0068 0035 DMNM    ,00111234Y
0069 0035 DMNM    ,00121259Y
0070 0036 D            1523                Y                P'***** GRAND TOTAL'
0071 0037 DTOTAL       1541Y J
0072 0038 D            1556                Y                P'*****'
0073 0039 D            2006                      Y          P'When you are finished -
```

Figure 11-9 (Part 3 of 8).   Generation Printout for Example Program 3

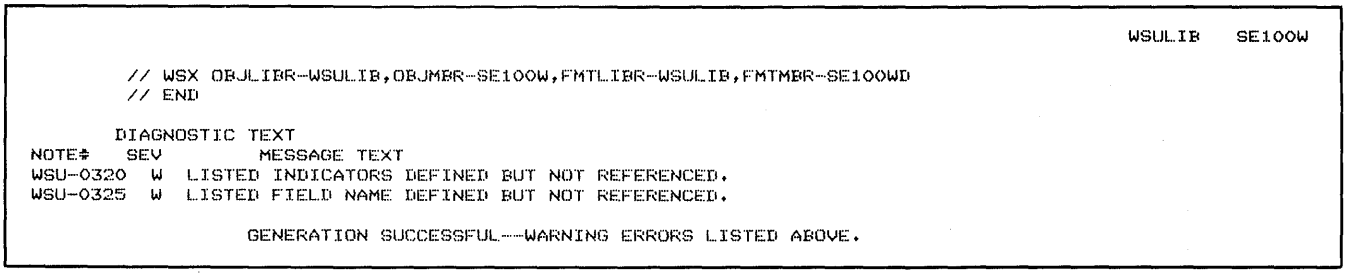


```
                                                                                           WSULIB   SE100W

       0040 Dviewing these totals, please press Enter'

       INDICATOR NAME USAGES                 RIGHT * = RECORD TYPE REFERENCE
         NAME       STMT#                    LEFT  * = DEFINITION
         EW         *0029
         IJ         *0018  *0019  *0020  *0021
         KG          0029   0030   0038   0039   0040   0041
         01         *0006  *0014   0036*  0036
         02         *0010   0041*  0041
         50         *0032   0033
         59         *0005
         69         *0013
         79         *0005
         89         *0013


       INDICATORS USED
         IJ EW KG 01 02 50

WSU-0320 UNREFERENCED INDICATORS
         59 69 79 89


       FIELD NAME AND LABEL USAGES           RIGHT * = UPDATE REFERENCE
         NAME       STMT#                    LEFT  * = DEFINITION
         *ERROR
         *RLNO
         *RLRN
         *RLRR
         *RLRU
         *SLNO
         *USID
         *WSID
         CODE       *0016
         DATE       *0008*  0025*  0031
         END         0030  *0037
         MNM        *0003  *0015   0058   0059   0060   0061   0062   0063   0064   0065   0066   0067   0068   0069
         MTH        *0031*  0032   0034   0034
         RCODE      *0007*  0035*
         RECNUM      0013  *0017   0018*
         SA         *0002   0021*  0034   0034*  0038   0040   0045   0046   0047   0048   0049   0050   0051   0052   0053
                     0054   0055   0056
         SALAMT     *0009*  0027*  0034
         SALTOT     *0004  *0012*  0021   0040*
         TCODE      *0011*  0039*
         TOTAL      *0038*  0071
         UDATE
         UDAY
         UMONTH
         UYEAR
```

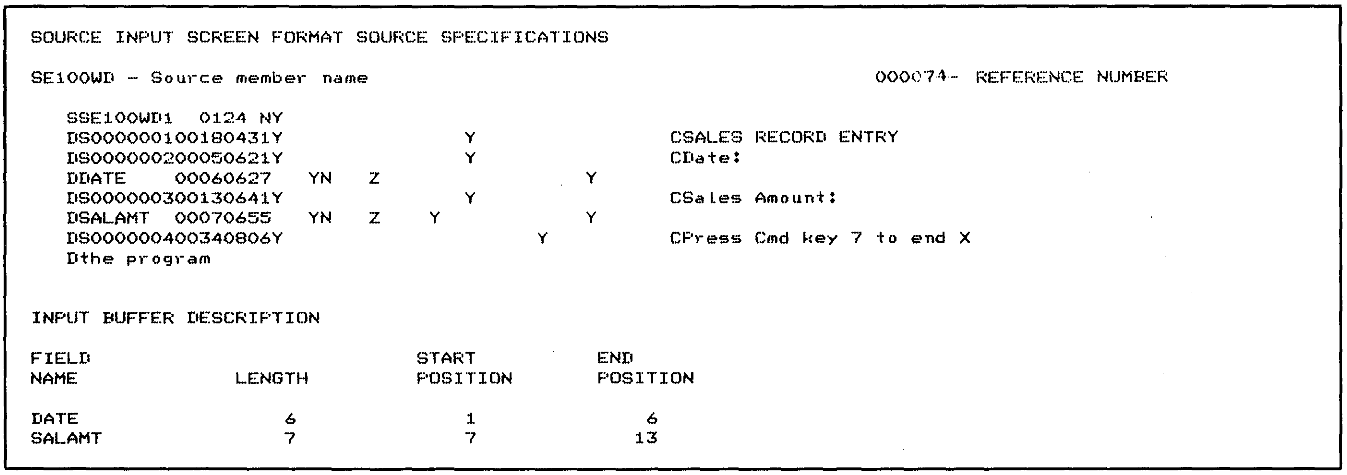
Figure 11-9 (Part 4 of 8). Generation Printout for Example Program 3

```
                                                                                           WSULIB   SE100W
       MODE LEVEL DATA FIELD NAMES USED
         NAME   DIM  STMT#   LNG    DEC    DISP  LCL F/ICNT SCR CHN
         SA     0012 0002    007     2     0000        000    D
         MNM    0012 0003    003     A     0054        001    D
         SALTOT 0012 0004    007     2     0078        001
         RCODE       0007    001     A     00CC        001
         DATE        0008    006     0     00CD        001    D
         SALAMT      0009    007     2     00D3        001    D
         TCODE       0011    001     A     00DA        001
         RECNUM      0017    003     0     00DB        001        M
         MTH         0031    002     0*    00DE        000
         TOTAL       0038    010     2     00E0        000    D

       PROGRAM LABEL NAMES USED
         NAME       STMT#    TYPE
         END        0037     TAG

WSU-0325 UNREFERENCED FIELD NAMES
         NAME       STMT#
         CODE       0016

       SOURCE/OBJECT CROSS REFERENCES
         STMT#    CSB  IAR  SSS     ****** DISPLAYED TEXT FOR PROGRAMMED HALTS (IMSG/MSG/DEBUG) ******
         0033     0007 002A 000A     INVALID DATE


       MAIN STORAGE REQUIREMENTS FOR WSU PROGRAM EXECUTION          MINIMUM        MAXIMUM
        BYTES REQUIRED PER WORK STATION                                  28            404
        TIMES MAXIMUM NUMBER OF WORK STATIONS                             1              1
        PLUS ADDITIONAL STORAGE REQUIRED                              10002          57368
        YIELDS STORAGE REQUIRED FOR EXECUTION                         10030          57772

        REGION REQUIRED FOR EXECUTION                                   10K            58K


       DISK STORAGE REQUIREMENTS FOR WSU EXECUTION PROGRAM WORKFILE
        SECTORS REQUIRED PER WORK STATION                                13
        TIMES MAXIMUM NUMBER OF WORK STATIONS                             1
        PLUS ADDITIONAL DISK SECTORS REQUIRED                             3
        YIELDS MINIMUM DISK SECTORS REQUIRED                             16

        EXECUTION WORKFILE SIZE IN BLOCKS                                 2


       SE100W   PROCEDURE CREATED FOR EXECUTION
        // ATTR NEP-NO,MRTMAX-1
        // REGION SIZE-10
        // LOAD #WSXI1
        // FILE NAME-SALESFLE,
        // IF DATAF1-SALESFLE DISP-OLD
        // ELSE DISP-NEW,RECORDS-100
        // FILE NAME-MONTHS,DISP-OLD
        // RUN
```

Figure 11-9 (Part 5 of 8). Generation Printout for Example Program 3



```
                                                                                           WSULIB   SE100W

        // WSX OBJLIBR-WSULIB,OBJMBR-SE100W,FMTLIBR-WSULIB,FMTMBR-SE100WD
        // END

       DIAGNOSTIC TEXT
NOTE#   SEV        MESSAGE TEXT
WSU-0320  W  LISTED INDICATORS DEFINED BUT NOT REFERENCED.
WSU-0325  W  LISTED FIELD NAME DEFINED BUT NOT REFERENCED.

                  GENERATION SUCCESSFUL---WARNING ERRORS LISTED ABOVE.
```



```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

SE100WD - Source member name                                          000074-  REFERENCE NUMBER

   SSE100WD1  0124 NY
   DS000000100180431Y                Y                CSALES RECORD ENTRY
   DS000000200050621Y                Y                CDate:
   DDATE     00060627   YN   Z                 Y
   DS000000300130641Y                Y                CSales Amount:
   DSALAMT   00070655   YN   Z    Y            Y
   DS000000400340806Y                      Y          CPress Cmd key 7 to end X
   Dthe program


INPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

DATE                 6             1              6
SALAMT               7             7             13
```

Figure 11-9 (Part 6 of 8). Generation Printout for Example Program 3

```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

SE100WD - Source member name                                   000074- REFERENCE NUMBER

   SSE100WD2  0124 NY
   DS000000500130435Y            Y              CSALES SUMMARY
   DSA     00100614Y
   DSA     00100639Y
   DSA     00100664Y
   DSA     00100814Y
   DSA     00100839Y
   DSA     00100864Y
   DSA     00101014Y
   DSA     00101039Y
   DSA     00101064Y
   DSA     00101214Y
   DSA     00101239Y
   DSA     00101264Y
   DMNM    00030609Y
   DMNM    00030634Y
   DMNM    00030659Y
   DMNM    00030809Y
   DMNM    00030834Y
   DMNM    00030859Y
   DMNM    00031009Y
   DMNM    00031034Y
   DMNM    00031059Y
   DMNM    00031209Y
   DMNM    00031234Y
   DMNM    00031259Y
   DS000000600171523Y            Y              C***** GRAND TOTAL
   DTOTAL  00141541Y
   DS000000700051556Y            Y              C*****
   DS000000800622006Y                  Y        CWhen you are finished vX
   Diewing these totals, please press Enter


EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                       START         END
NAME           LENGTH       POSITION      POSITION

SA                10            1            10
SA                10           11            20
SA                10           21            30
SA                10           31            40
SA                10           41            50
SA                10           51            60
SA                10           61            70
SA                10           71            80
SA                10           81            90
SA                10           91           100
SA                10          101           110
SA                10          111           120
MNM                3          121           123
```

Figure 11-9 (Part 7 of 8). Generation Printout for Example Program 3

```
MNM                3             124            126
MNM                3             127            129
MNM                3             130            132
MNM                3             133            135
MNM                3             136            138
MNM                3             139            141
MNM                3             142            144
MNM                3             145            147
MNM                3             148            150
MNM                3             151            153
MNM                3             154            156
TOTAL             14             157            170
```

```
  WSULIB  - INPUT LIBRARY NAME

  WSULIB  - Output library name

  SE100WD - Format load member name                                 000074-  REFERENCE NUMBER

FORMAT SE100WD1 REQUIRES   256 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   122 BYTES
FORMAT SE100WD2 REQUIRES   768 BYTES OF STORAGE AND HAS A DATA STREAM LENGTH OF   368 BYTES
```

Figure 11-9 (Part 8 of 8). Generation Printout for Example Program 3

Chapter 12. F-(File Description) and I-(Input) Specifications

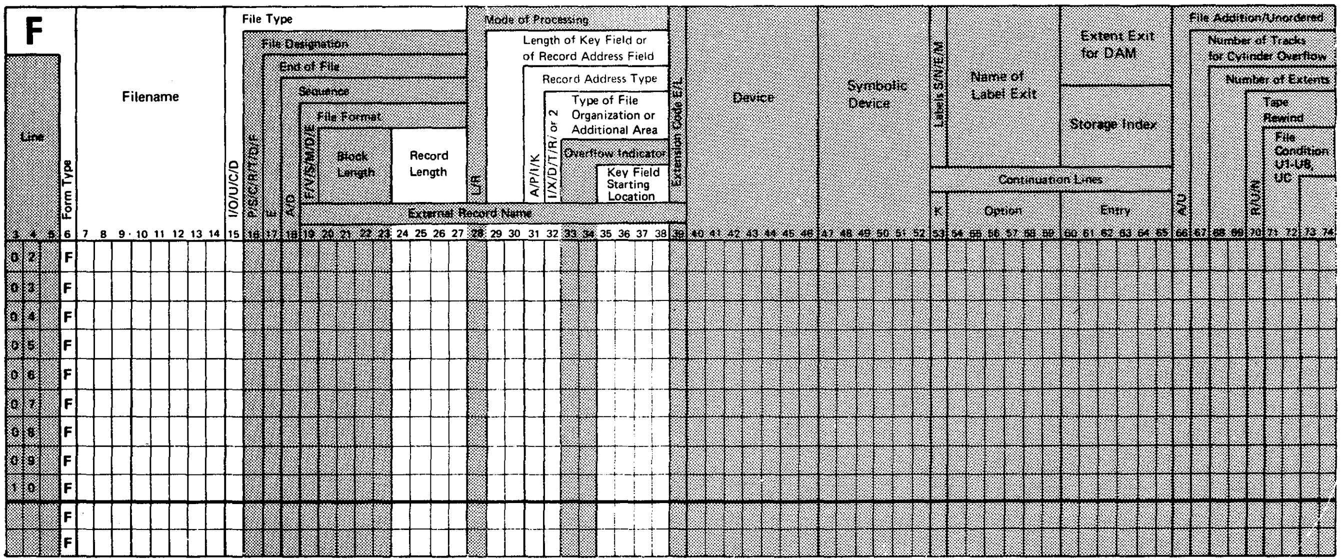
This chapter summarizes the entries on the F- and I-specifications. A description and examples of the entries used by WSU can be found in *Coding Files* in Chapter 4. Example programs using F- and I-specifications can be found in Chapter 11, *WSU Example Programs*.

F-Specification

The file description specifications, which are coded on the *RPG Control and File Description Specifications* sheet, are RPG specifications that are used in your WSU program. The F-specifications are entered before your I-specifications in your file definition. WSU just uses the information from these specifications that is needed to run your WSU program. Columns 16 through 23, 28, 33 and 34, and 39 through 80 are not used by WSU; entries in these areas are ignored. The top area of this sheet, the control specifications, are not used by WSU.

File Description Specifications

For the valid entries for a system, refer to the RPG reference manual for that system.



*Number of sheets per pad may vary slightly.

The following entries on the F-specifications are used by WSU:

File Name (Columns 7 through 14)

The *file name* is a required entry and specifies the name of the transaction or master file used by the WSU program. No two files in the program can have the same name. The file name can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded blanks are not allowed.

File Type (Column 15)

Even though WSU treats the transaction file as a direct output file, code either **I** (input file) or **U** (update file) for the *file type*. Should these same specifications be used in an RPG II program, the I or U will have meaning to that program. A master file is an update file if a PUT or PUTN operation occurs for the file; it is an input file if a PUT operation does not occur for the file.

If column 15 is **blank** or if any **other entry** is made, WSU continues processing but issues a terminal-error message for the file type, and does not produce a program because of the error.

Record Length (Columns 24 through 27)

The length (in bytes) of the records in the transaction file can be any number from **14 through 4096**. This entry must be right-adjusted; leading blanks or zeros are allowed. The record length for a transaction file must include 13 bytes for the trailer; however, data cannot be specified in these last 13 bytes of the records.

The length (in bytes) of the records in the master file can be any number from **1 through 4096**. This entry must be right-adjusted; leading blanks or zeros are allowed.

If columns 24 through 27 are left **blank**, WSU assumes a 256-byte record length, and continues processing the specifications, but issues a terminal-error message for the blank record length, and does not produce a program because of the error.

Length of Key Field (Columns 29 and 30)

This entry applies only to indexed master files. Columns 29 and 30 must be blank for a direct file.

The length (in bytes) of the record key can be any number from **1 through n**. The key length must be right-adjusted, and leading blanks and zeros are allowed.

- For a packed key, the maximum length is 8.
- For a non-contiguous key, the maximum length is 99. (The length is the sum of the 2 or 3 fields used to make up the non-contiguous key.)
- For all other keys, the maximum length is 99.

For a **blank** or any **other entry** WSU assumes 3, and continues processing, but issues a terminal-error message and does not produce a program.

Record Address Type (Column 31)

This entry applies only to indexed master files.

An entry of **P** signifies that the key is a packed decimal record key. An entry of **A** signifies that the entry is an alphameric record key. WSU assumes an A if a **blank** or any **other entry** is made.

See Chapter 4, *Coding Files*, for a discussion of packed decimal and alphameric record keys.

Type of File Organization (Column 32)

This entry applies only to master files.

An entry of **I** specifies that the type of master file is an indexed master file. If a **blank** or any **other entry** is made, WSU assumes a direct file organization.

See Chapter 4, *Coding Files*, for a discussion of file organization.

Key Field Starting Location (Columns 35 through 38)

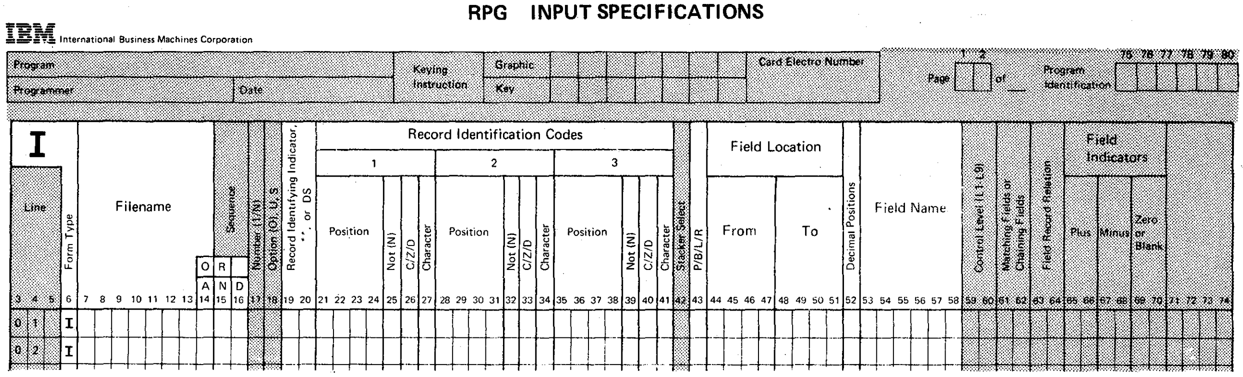
This entry applies only to indexed master files; it must be blank for a direct file. The starting position of a contiguous key in each record of an indexed master file can be any number from **1 through n**. The entry must be right-adjusted; leading blanks or zeros are allowed. The maximum value (n) is the record length. The key's starting location plus the key's length minus one cannot exceed the record length.

For a non-contiguous key, **EXTK** is entered in columns 35 to 38 instead of the key field starting location.

If a **blank** or any **other entry** is made, WSU assumes 1, continues processing, but issues a terminal-error message for the invalid starting location, and does not produce a program because of the error.

I-Specification

Input specifications are coded on the *RPG Input Specifications* sheet and follow the F-specifications in your file definition. WSU just uses I-specification entries needed to describe your files. The shaded areas are reserved; entries in these areas are ignored by WSU.



The following entries on the I-specifications are used by WSU:

File Name (Columns 7 through 14)

The *file name* is a required entry and specifies the name of a file or data area. No two files in the program can have the same name. The file name can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded blanks are not allowed.

An entry in *File Name*:

- Must be the same name that is in *File Name* (columns 7 through 14) of an F-specification (unless the I-specifications describe session-level fields or local-data-area fields)
- Must be the same name that is in *File Name* (columns 7 through 14) of an M- or T-specification
- Must come before the associated field lines
- Can be omitted when multiple record types are defined for the same file, but must be coded for the first record type. (This restriction does not apply to I-specifications that describe session-level fields or local-data-area fields).

AND/OR (Columns 14 through 16)

The AND/OR lines are used to indicate a relationship between record IDs. For a further explanation of this entry, see the description of *Record Identification Codes* (columns 21 through 41) and *Field Name* (columns 53 through 58) of the I-specification in this chapter.

WSU ignores entries other than AND or OR. For example, if you are using I-specifications from an RPG II program, or if you code the I-specifications so that a follow-on RPG II program can use them, columns 15 and 16 can contain characters or numbers that sequence record types.

Note: For coding information about AND/OR, see Chapter 4, *Coding Files*.

Record Identifying Indicator (Columns 19 and 20)

The *record identifying indicator* can be any number from **01 through 89**. This 2-digit entry assigns an indicator to a record type. Record identifying indicators must be unique within a file.

The record identifying indicator entry *must be* blank on a continued I-specification.

Record Identification Codes (Columns 21 through 41)

These entries describe a record type. Each I-specification record statement contains one, two, or three sets of record identification codes, each with the following subentries:

- Position (columns 21 through 24, 28 through 31, and 35 through 38)
- Not (columns 25, 32, and 39)
- C/Z/D (columns 26, 33, and 40)
- Character (columns 27, 34, and 41).

The last set of record identification codes should be the last information coded on that line (columns 42 through 80 should be blank).

Position (Columns 21 through 24, 28 through 31, 35 through 38)

The location in the record of the record identification character can be any number **1 through n**. Entries must be numeric and right-adjusted. The maximum allowed position, *n*, is the length of the records for a master file (4096 is the maximum) and the length of the records minus 13 for the transaction file (4083 is the maximum). The 13 positions are reserved for the trailer. If this entry exceeds the maximum length, WSU assumes that the record identification character is the last character in the record.

A **blank** in these columns means that a record identification character is not specified.

Not (Columns 25, 32, 39)

An **N** specifies that the position in the record does *not* contain the character in column 27, 34, or 41. A **blank** specifies that the position in the record does contain the character in column 27, 34, or 41. If any **other entry** is made, WSU assumes N.

C/Z/D (Columns 26, 33, 40)

This entry indicates which portion of the character in column 27, 34, or 41 is used as the record identification character. Many characters have either the same zone portion or digit portion.

C	The entire character specified in column 27, 34, or 41 is the record identification character.
Z	The zone portion of the character specified in column 27, 34, or 41 is the record identification character.
D	The digit portion of the character specified in column 27, 34, or 41 is the record identification character.
Other	Assume C.

Refer to the *Sort Guide* for more about choosing which portion of a character to use as the record identification character.

Character (Columns 27, 34, 41)

This entry specifies an alphabetic character, special character, or digit as the record identification character.

When selecting characters for record identification by a digit or zone only, WSU selects all characters having the same zone or digit. When reading characters, WSU converts each character into an 8-bit code and then tests this 8-bit code to see if the character meets the requirements of the record identification character in the I-specification.

As an example, a digit-only entry (D) in column 26 and an A in column 27 cause WSU to select all records with a / (slash), A, a, J, j, or 1 in the specified column. Using the same letter A but now selecting records on a zone-only basis (a Z in column 26), WSU selects all records with & and letters A through I in the specified column.

P/B/L/R (Column 43)

This entry indicates whether a numeric field is in packed decimal or binary format.

Blank	Zoned decimal numeric field or alphameric field.
P	Packed decimal numeric field. The maximum length is 8 bytes.
B	Binary field. The field must have a length of either 2 or 4 bytes.
L/R	WSU issues a warning message, and does no further checking of this specification's entries.
Other	WSU issues a terminal-error message and assumes blank.

If P or B is specified for a local-data-area field or a session-level field, WSU issues a terminal-error message and assumes blank.

Field Location (Columns 44 through 51)

This entry has two subentries:

- *From* (columns 44 through 47) contains the position in which the field named in columns 53 through 58 begins.
- *To* (columns 48 through 51) contains the position in which the field named in columns 53 through 58 ends.

The From and To entries must be right-adjusted. The From entry must be less than or equal to the To entry. If the From or To entry is not numeric or is not right-adjusted, WSU assumes 1 for both entries.

The starting position of the field, *From* (columns 44 through 47), and the ending position of the field, *To* (columns 48 through 51), can be any number from 1 through n. For a master file, the maximum entry, *n,* is the length of the record. For the transaction file, fields can neither begin nor end in the last 13 bytes of the record (these 13 bytes are reserved for the trailer). For local-data-area fields and session-level fields, *n* is 256. For session-level fields, this entry is used by WSU only to determine the field length.

For a master file, if the To entry exceeds the record length, WSU assumes that the To entry is the record length. For the transaction file, if the To entry exceeds the record length minus 13, WSU assumes that the To entry is the record length minus 13 (13 positions are reserved for the trailer).

Maximum field lengths are as follows:

- 256 for an alphameric field. If the field is longer than 256 characters, WSU assumes 256.
- 15 for a zoned decimal numeric field. If the field is longer than 15 digits, WSU assumes 15.
- 8 for a packed numeric field. If the field is longer than 8 digits, WSU assumes 8.
- 4 for a binary field. If the field is longer than 4 bytes, WSU assumes 2.

Note: The field location entry (columns 44 through 51) is required for a field statement on an I-specification.

Decimal Positions (Column 52)

This entry indicates the number of positions to the right of the decimal point in a numeric field named in columns 53 through 58; it can be any number from **0 through 9**. Column 52 must contain an entry when the field named in columns 53 through 58 is numeric. To define a numeric field with no decimal positions, column 52 must be 0. If the column is left **blank**, WSU assumes that the field is an alphameric field. If any entry other than 0 through 9 is made, WSU assumes 0.

Fields that WSU edits or uses in arithmetic operations must be numeric. If the number of decimal positions exceeds the field length, WSU assumes that the number of decimal positions is the same as the field length. WSU cannot process numbers that have more than nine decimal places.

Note: Column 52 must not be blank for a packed or binary field.

Field Name (Columns 53 through 58)

This entry specifies the name of a field. *Field names* can be duplicates of other field names in the program's file definition if the length and number of decimal positions match.

A field name can be from **1 to 6 characters** long and must be left-adjusted. The first character must be alphabetic; the remaining characters can be alphameric. Embedded blanks are not allowed. Reserved field names cannot be used. (Refer to Chapter 3, *Reserved Fields and Indicators*, for a list of reserved field names.)

A job field can be defined in columns 53 through 58. Only one copy of a job field is generated per program, rather than one copy per each display station that uses the program. Job field names must begin with & and must be left-adjusted. The second character must be alphabetic; the remaining characters (up to 4) must be alphameric. Job field names cannot contain blanks.

Note: The field name is required in columns 53 through 58 for a field statement on an I-specification.

Chapter 13. J-(Job) Specification

This chapter summarizes the J-(Job) specification entries. Using J-specifications in sample programs is illustrated in Chapter 11, *WSU Example Programs*.

The J-specification is coded on the top area of the *WSU Job, Array, and File Specifications* sheet. The shaded areas are reserved; entries in these areas are ignored by WSU. When entering the WSU specifications on the System/36, the J-specification is the first specification entered.

Job Specification

Sequence Number 1 2 3 4 5	Specification Type 6	Program Name 7 8 9 10 11 12 13 14	Display Format Member Name 15 16 17 18 19 20 21 22	Message Load Member Name 23 24 25 26 27 28 29 30	Library Name 31 32 33 34 35 36 37 38	Maximum Number of Display Stations 39 40	Region Size 41 42	Date/Edit 43	Ideographic Mode 44	**Second Edition** Use this coding sheet only with WSU programs. This coding sheet could contain typographical errors. Reserved 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	J									▨

The following entries on the J-specification are used by WSU:

Program Name (Columns 7 through 14)

The *program name* is a required entry and specifies the name that is to be assigned to the WSU program *and* to the WSU-generated procedure. The program name can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), hyphens, and periods are not allowed.

Display Format Member Name (Columns 15 through 22)

The *display format member name* is a required entry and specifies the name of the load member in which generated display formats are to be stored. The display format member name can be from **1 to 8 characters** and must be left-adjusted. The first character must alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), hyphens, and periods are not allowed.

The display format member name must not be the same as the message load member name in columns 23 through 30. Also, the display format member name must not be the same as the WSU source program name when the display format member and the source program are in the same library.

Message Load Member Name (Columns 23 through 30)

The *message load member name* is an optional entry and specifies the name of your message load member that contains any messages you reference by MIC (message identification code) on a D-specification or a C-specification. If specified, the message member name can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), hyphens, and periods are not allowed.

The message load member name cannot be the same as the display format member name (columns 15 through 22).

If the message load member name is **blank**, WSU assumes that, if user messages are issued in your program, either you or the operator will modify the WSU-generated procedure and specify the message member to use before calling the program. Refer to *Modifying a WSU-Generated Procedure* in Chapter 8 for a description of how to change the user procedure. Refer to the *System Reference* manual for an explanation of how to create and compile a message member.

Library Name (Columns 31 through 38)

This entry indicates either:

- The name of the library that will contain the output produced during WSU program generation.
- The name of the library that contains the program's user message load member, if columns 23 through 30 of the J-specification contain a message load member name.

The *library name* can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), and hyphens are not allowed.

If columns 31 through 38 are **blank**, the current library is used for generating output, and WSU looks for the message load member in the current library.

Maximum Number of Display Stations (Columns 39 and 40)

This entry specifies the *maximum number of display stations* that can use the program at the same time.

Any right-adjusted number from **1 through 99** can be entered. If a **blank** or any **other character** is entered, WSU assumes 1.

If your program should allow more than 99 display stations or if for other reasons you want to modify the maximum number of display stations, you can change the value of the MRTMAX parameter on the ATTR OCL statement in the generated WSU procedure. Refer to *Modifying a WSU-Generated Procedure* in Chapter 8 for an explanation of how to change the procedure.

Region Size (Columns 41 and 42)

This entry specifies the amount of main storage (in additions of 2K bytes) in which your program runs. (The region size does not affect the minimum size of the program that WSU generates.)

This entry should be an even number from **8 through 64** and must be right-adjusted. If you code an odd number, WSU uses the next higher even number as the region size. If a **blank** or any **other entry** is specified, WSU will assume the minimum region size in which the program can run.

If the region size specified is less than the minimum region size in which the program can run, the minimum region size is used.

The minimum region size in which the program can run is printed at the end of the generation printout. Refer to Chapter 9, *Testing and Debugging WSU Programs*, for an explanation of output printed during generation.

You can modify the region size in the WSU-generated procedure before calling the procedure. Refer to *Modifying a WSU-Generated Procedure* in Chapter 8 for further information.

Date/Edit (Column 43)

This entry, together with the edit code specified on the D-specification, determines the format of the program date or numeric output fields.

M The format of the program date is month (MM), day (DD), year (YY); and a Y edit code in column 25 of a D-specification edits these dates as MM/DD/YY. This entry, along with a J edit code in column 25 of a D-specification, edits a numeric output field as follows:

1,234.56ƀ	(Format of a positive number)
1,234.56-	(Format of a negative number)
0.00	(Format of a zero balance)

D The format of the program date is day (DD), month (MM), year (YY); and a Y edit code in column 25 of a D-specification edits these dates as DD/MM/YY.

This entry, along with a J edit code in column 25 of a D-specification, edits a numeric output field as follows:

1.234,56ƀ	(Format of a positive number)
1.234,56-	(Format of a negative number)
0,00	(Format of a zero balance)

Y The format of the program date is year (YY), month (MM), day (DD); and a Y edit code in column 25 of a D-specification edits these dates as YY/MM/DD.

This entry, along with a J edit code in column 25 of a D-specification, edits a numeric output field as follows:

1.234,56ƀ	(Format of a positive number)
1.234,56-	(Format of a negative number)

Other Assume M.

Code numeric literals on C-specifications or D-specifications according to the edit code you specify for this entry.

Refer to *Edit Code (Column 25)* in Chapter 16 for examples of editing fields.

Ideographic Mode (Column 44)

Y This WSU program can be run only from an ideographic display that is in ideographic mode. A display station is in an ideographic session if Y is entered for the ideographic (IGC) session prompt on the sign on display.

N This WSU program cannot be run from a display that is in ideographic mode.

Blank Assume N.

Chapter 14. E-(Array) Specifications

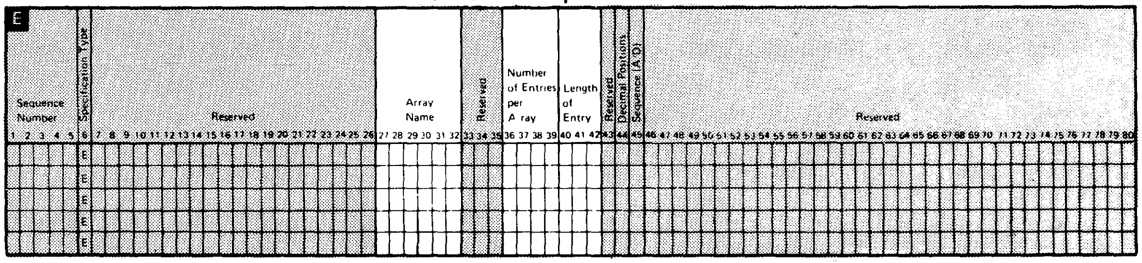
Array specifications used in a WSU program should be written on the E-specifications (middle area) of the *WSU Job, Array, and File Specifications* sheet. Columns 7 through 26, 33 through 35, and 43 are not used by WSU, and an error message is issued if there are entries in these columns. Columns 46 through 80 are ignored by WSU and no error message is issued.

When entering the WSU specifications on the System/36, the E-specifications must be entered after the J-specification (Chapter 13) and before the T- and M-specifications.

For more about arrays in WSU, see:

- Chapter 5 *Coding Arrays*
- Chapter 6 *Coding Displays*
- Chapter 16 *S-(Display Control)* and *D-(Field Definition) Specifications.*

Transaction/Master File Specifications



The following entries on the E-specification are used by WSU:

Array Name (Columns 27 through 32)

This entry indicates the name of the array used in the program.

Each array used in a program must be given a unique name that does not begin with the letters TAB. (TAB is used for RPG tables and should not be used in WSU arrays). The name can be from **1 to 6 characters** long and must begin with an alphabetic character. This array name is used throughout the program. The array name should be used by itself only to reference the entire array.

Number of Entries in the Array (Columns 36 through 39)

Columns 36 through 39 are used to indicate the maximum number of elements that can be contained in the array named in columns 27 through 32. The maximum number of elements that can be used by WSU can be any number from **1 through 9999**. If this entry is left blank, WSU assumes the number to be 5. This entry must be right-adjusted.

Length of Entry (Columns 40 through 42)

The length of an *alphameric* element for WSU arrays can be any number from **1 through 256**. This entry must be right-adjusted. If this entry is left blank, WSU assumes a length of 5.

The length of a *numeric* array element can be any number from **1 through 15**. If this entry is left blank, WSU assumes a length of 15. The number entered must be right-adjusted in column 42. For numeric arrays in packed decimal format, enter the zoned decimal length in columns 40 through 42. For numeric arrays in *binary* format, enter the number of digits required in storage for the binary field. For a 2-position binary field, the entry in columns 40 through 42 is 4; for a 4-position binary field, the entry is 9.

Decimal Positions (Column 44)

Column 44 is used to indicate the number of decimal positions in a numeric array element; **0 through 9** positions are allowed. Column 44 must always have a value for a numeric array. If the elements in a numeric array have no decimal positions, enter a 0.

If you are using alphameric arrays, column 44 must be left **blank**.

Sequence A/D (Column 45)

Use column 45 to describe the *sequence*, either ascending or descending, of the data in an array. WSU arrays are not sequence-checked; however, an **A** or **D** entry must be specified if a high or low LOKUP operation is performed. See *LOKUP* in Chapter 18 for more information.

An *A* entered for ascending order means that the array entries start with the lowest data entry (according to the collating sequence) and proceed to the highest. A *D* entered for descending order means that the array entries start with the highest data entry and proceed to the lowest.

If the sequence entry is left **blank**, the entry is assumed to be an A.

Refer to the *Sort Guide* for more about the standard collating sequence.

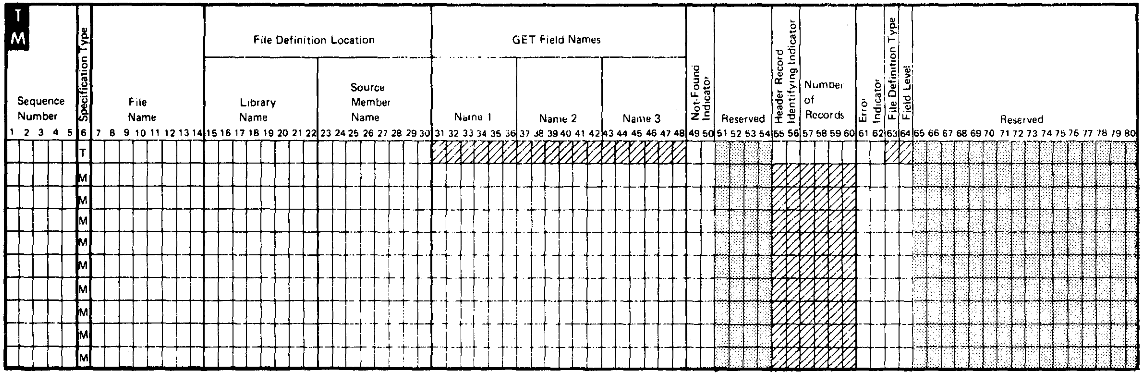
Chapter 15. T-(Transaction File) and M-(Master File) Specifications

This chapter summarizes the entries on the transaction file and master file specifications. These specifications are coded on the lower area of the *WSU Job, Array, and File Specifications* sheet. When entering the WSU specifications on the System/36, the E-specifications (Chapter 14) must be entered after the J-specifications (Chapter 13) and before the T- and M-specifications.

T-Specification

The transaction file specification is entered on the top line of the Transaction/Master File section of the *WSU Job, Array, and File Specifications* sheet, and must follow the E-specifications (if any) in your WSU program. Columns 31 through 48 and 51 through 54 are not used for transaction files, and an error message is issued if there are entries in these columns. Columns 63 through 80 are reserved; entries in these columns are ignored by WSU.

Transaction/Master File Specifications



The following entries on the T-specification are used by WSU:

File Name (Columns 7 through 14)

The *file name* is a required entry and specifies the name of the transaction file. The file name can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), hyphens, and periods are not allowed. The transaction file name cannot be the same as any of the master file names in this program. The transaction file name also appears in columns 7 through 14 of the F-specification and I-specification for that file.

File Definition Location (Columns 15 through 30)

The *file definition location* contains two subentries for the transaction file:

- Library Name (columns 15 through 22)
- Source Member Name (columns 23 through 30).

File Definition Location (Columns 15 through 30)

The *file definition location* contains two subentries for the transaction file:

- Library Name (columns 15 through 22)
- Source Member Name (columns 23 through 30).

Library Name (Columns 15 through 22)

This entry indicates the name of the library that contains the F-specification and I-specifications (file definition) for the transaction file.

The *library name* can be from **1 to 8 characters**, and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric (including periods). Embedded commas, single quotation marks, blanks, question marks, slashes (/), and hyphens are not allowed. If columns 15 through 22 are **blank**, the current library is used.

Source Member Name (Columns 23 through 30)

This required entry indicates the name of the source member that contains the F-specification and I-specifications (file definition) for the transaction file. The file definition that describes the transaction file must not be in the same source member that contains the WSU program.

The *source member name* can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), hyphens, and periods are not allowed.

Not-Found Indicator (Columns 49 and 50)

An entry of an indicator number from **01 to 89** specifies the indicator that is turned on when:

- WSU cannot determine the record type of the record read from the transaction file.
- WSU has tried to get a record that is beyond the logical end of the transaction file chain for that display station.

You can also code a not-found indicator in columns 54 and 55 of C-specifications for the GETNH, GETNR, GETPH, and GETPR operations. If you code different not-found indicators on the T-specification and C-specification, the indicator on the C-specification turns on for a not-found condition.

This indicator remains on until:

- You turn it off (with a SETOF operation).
- WSU turns it off (when the indicator, used as a resulting indicator in columns 54 through 59 of the C-specification, reflects a not-true condition or when an allowed record is read).

Header Record Identifying Indicator (Columns 55 and 56)

This indicator number can be any number from **01 through 89** but must be the same as the record identifying indicator coded on the I-specification for the header record in the transaction file. Code this indicator if the transaction file consists of header records followed by groups of detail records (for example, customer orders in an order-entry application). This entry allows operators to use the Page Backward Group and Page Forward Group command keys during review mode to page from one header record to the next, and allows you to code GETNH and GETPH operations.

If columns 55 and 56 are **blank**, WSU assumes no grouping by header record and detail records. The Page Backward Group command key and Page Forward Group command key are not allowed. If pressed, either key causes an error message to appear for the operator. Also, either the GETPH or GETNH operation causes a terminal error to occur.

Note: Refer to Chapter 10, *ATTENTION: OPERATORS (Running a WSU Program)*, for a description of how the Page Backward Group and Page Forward Group command keys work. Refer to Chapter 18 for a description of the GETNH and GETPH operations.

Number of Records (Columns 57 through 60)

Use any number from **1 through 9999** to specify the number of records that operators can enter in the transaction file. WSU uses the number of records to allocate sufficient disk storage for the transaction file. The entry must be right-adjusted.

If the number of records is left **blank**, WSU allocates disk storage for 1000 records for the transaction file.

Notes:

1. Disk space is allocated in blocks of 2560 characters. For this reason, more space for records may be allocated than you requested.
2. If you need more records than the 9999 that you can specify on the T-specification, you can use SEU to modify the WSU-generated procedure and increase the number of records up to a maximum of 65,535. Refer to Chapter 8, *Entering and Generating a WSU Program*, for information on how to modify the procedure.

Error Indicator (Columns 61 and 62)

An entry of an indicator number from **01 through 89** specifies the indicator that is turned on when either of the following conditions occurs:

- An output/input error occurs for a transaction file operation (GETNH, GETNR, GETPH, GETPR, or PUT)
- A record-not-found condition occurs for the transaction file and an indicator has not been coded for Not-Found Indicator in columns 49 and 50 of the T-specification.

A code for the error that occurred is placed in *ERROR, a WSU reserved field. (Refer to Chapter 3 for a description of *ERROR and the possible error codes). If an error does not occur, this indicator is not set on.

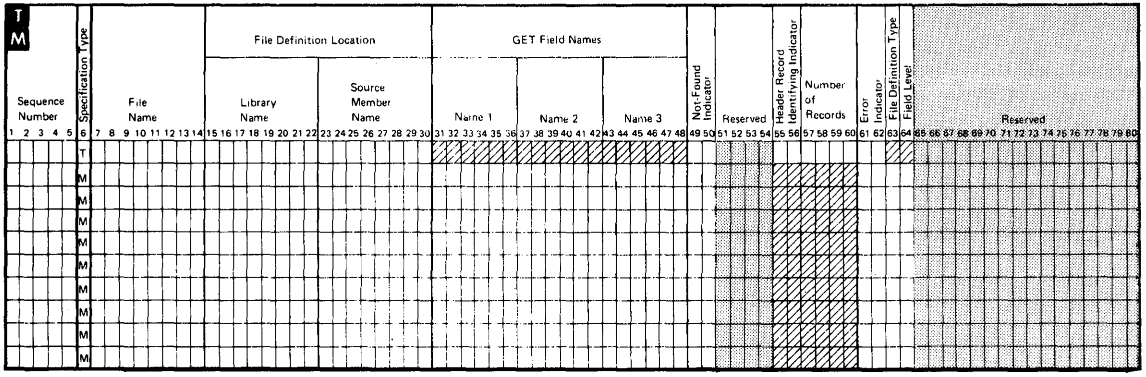
If these columns are left **blank**, WSU assumes that no error indicator is specified for the transaction file.

If any **other entry** is made, WSU issues a warning message during program generation and assumes blank.

M-SPECIFICATION

The master file specifications are coded on the lines following the T-specification on the *WSU Job, Array, and File Specifications* sheet. When entering the WSU specifications on the System/36, the M-specification must be entered after the T-specification and before the S-specification (Chapter 16). Columns 65 through 80 are reserved; entries in these areas are ignored. Columns 51 through 60 are not used by WSU; an error message will be issued if there are entries in these columns.

Transaction/Master File Specifications



T M Sequence Number 1 2 3 4 5	Specification Type 6	File Name 7 8 9 10 11 12 13 14	File Definition Location Library Name 15 16 17 18 19 20 21 22	File Definition Location Source Member Name 23 24 25 26 27 28 29 30	GET Field Names Name 1 31 32 33 34 35 36	GET Field Names Name 2 37 38 39 40 41 42	GET Field Names Name 3 43 44 45 46 47 48	Not-Found Indicator 49 50	Reserved 51 52 53 54	Header Record Identifying Indicator 55 56	Number of Records 57 58 59 60	Error Indicator 61 62	File Definition Type 63	Field Level 64	Reserved 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	T														
	M														
	M														
	M														
	M														
	M														
	M														
	M														
	M														
	M														

WSU uses the following entries on the M-specification:

File Name (Columns 7 through 14)

This entry is the name of a master file, a group of session-level fields, or a local data area.

The *file name* can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), hyphens, and periods are not allowed.

The master file name cannot be the same as the transaction file name or another master file in this program. The master file name also appears in columns 7 through 14 of the F-specification and I-specification for that file.

The name of session-level fields or the name of a local data area also is specified in columns 7 through 14 of an I-specification in the program's file definition.

File Definition Location (Columns 15 through 30)

The *file definition location* for the master file contains two subentries:

- Library Name (columns 15 through 23)
- Source Member Name (columns 24 through 30).

Library Name (Columns 15 through 22)

This entry indicates the name of the library that contains the F-specification and I-specifications (file definition) for the master file.

The *library name* can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric (including periods). Embedded commas, single quotation marks, blanks, question marks, slashes (/), and hyphens are not allowed. If columns 15 through 22 are **blank**, the current library is assumed.

Source Member Name (Columns 23 through 30)

This entry indicates the name of the file definition source member that contains:

- F- and I-specifications that describe the master file named in columns 7 through 14.
- I-specifications that describe session-level fields.
- I-specifications that describe local-data-area fields.

The file definition must not be in the same source member as the WSU program.

The *source member name* can be from **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. Embedded commas, single quotation marks, blanks, question marks, slashes (/), hyphens, and periods are not allowed.

GET Field Names (Columns 31 through 48)

For GET operations to a master file, WSU forms a single field by combining the GET fields (in the order coded on the M-specification) and then uses the newly formed field to get records in the master file.

These entries specify from one to three fields that, when placed end-to-end, contain a value that corresponds to the key field of the indexed master file or the relative record number of a direct master file. An M-specification requires at least one GET field name in columns 31 through 48. Each name must be left-adjusted. The GET fields must have been defined on I-specifications or C-specifications in the program.

For indexed files, the combined length of the GET fields must be the same as the length of the key field in the master file.

Direct files and files with packed keys can have only one GET field. The GET field name for direct files must be 1 to 6 characters in length and must be numeric.

Not-Found Indicator (Columns 49 and 50)

An indicator number entry from **01 through 89** specifies the indicator that is turned on when:

- WSU cannot determine the record type of the record read from the master file.
- There is no record that has the specified key.

You can also code a not-found indicator in columns 54 and 55 of C-specifications for a GET operation to a master file. If you code different not-found indicators for the same file on the M-specification and C-specification, the indicator on the C-specification turns on for a not-found condition.

This indicator remains on until:

- You turn it off (with a SETOF operation)
- WSU turns it off (when the indicator, used as a resulting indicator in columns 54 through 59 of the C-specification, reflects a not-true condition or when an allowed record is read).

Error Indicator (Columns 61 and 62)

An indicator number entry from **01 through 89** specifies the indicator that turns on when either of the following conditions occurs:

- An output/input error occurs for a master file operation (GET, PUT, or PUTN)
- A record-not-found condition occurs for the master file and an indicator has not been coded in columns 49 and 50 (Not-Found Indicator) of the M-specification.

A code for the error that occurred is placed in *ERROR, a WSU reserved field. (Refer to Chapter 13 for a description of *ERROR and the possible error codes.) If an error does not occur, this indicator is not set on.

If the error indicator is left **blank**, WSU assumes that no error indicator is specified for the master file.

If any **other entry** is made, WSU issues a warning message during program generation and assumes blank.

File Definition Type (Column 63)

These columns describe the fields, master file, or local data area named in columns 7 through 14 of the M-specification.

If this entry is left **blank**, WSU assumes the use of a *master file*. Columns 7 through 14 of this M-specification name the master file and columns 23 through 30 name the source member that contain the file definition F-specification and I-specifications that describe the master file.

An entry of **U** specifies the use of the *local data area*. Columns 7 through 14 of this M-specification name the local data area, and columns 23 through 30 name the source member that contains the I-specifications that describe the local data area.

An entry of **F** specifies the use of *fields* by the WSU program rather than use of a master file or of the local data area. Columns 7 through 14 of this M-specification name the group of fields, and columns 23 through 30 name the source member that contains the I-specifications that describe the fields. Field Level (column 64) specifies whether the fields named in file definition type are mode-level or session-level fields.

If any **other entry** is made, WSU issues a terminal-error message and does not generate a program because of the error.

Figure 15-1 shows possible combinations of entries in columns 63 and 64.

	Column 64	
Column 63	**Blank**	**S**
Blank	Master file fields. These fields are mode-level.	Invalid combination. WSU issues a terminal error message.
U	Local-data-area fields. These fields are mode-level.	Local-data-area fields. These fields are session-level.
F	Invalid combination. WSU issues a terminal error message.	Session-level fields.

Figure 15-1. Possible Combinations of File Definition Types and Field Levels

Field Level (Column 64)

This entry indicates whether the fields named in columns 7 through 14 and identified in column 63 are session-level or mode-level.

An entry of **S** in column 64 specifies one of the following:

- If the file definition type in column 63 is U, the local-data-area fields named by this M-specification are session-level.
- If the file definition type in column 63 is F, the group of fields named by this M-specification are session-level.
- If the file definition type in column 63 is blank, WSU issues a terminal-error message and does not generate a program because of the error.

A **blank** in column 64 specifies one of the following:

- If the file definition type in column 63 is U, the local-data-area fields named by this M-specification are mode-level.
- If the file definition type in column 63 is F, WSU issues a terminal-error message and does not generate a program because of the error.
- If the file definition type in column 63 is blank, the fields in the master file named by this M-specification are mode-level fields.

If any **other entry** is made, WSU issues a terminal-error message and does not generate a program because of the error.

Note: Figure 15-1 shows possible combinations of entries in columns 63 and 64.

Chapter 16. S-(Display Control) and D-(Field Definition) Specifications

This chapter summarizes the entries for WSU on the *System/36 Display Format Specifications* sheet. A description and examples of these entries can be found in Chapter 6, *Coding Displays*. Sample programs using S- and D-specifications can be found in Chapter 11, *WSU Example Programs*.

S-Specification

S-(display control) specifications are coded on the top section of the sheet. When entering the WSU specifications for your program, you must enter the S-specification after the M-specifications (Chapter 15) and before the D-specifications. Entries in columns 31 through 36 will cause WSU to issue a warning message and continue as though those columns were blank. Columns 39, 40, 45, and 60 through 63 are reserved and entries in these columns are ignored.

Display Control Specification

S																			WSU Only															
																			Enter Mode Sequence							Review Mode Record Identifying Indicators			Insert Mode Record Identifying Indicators					
Sequence Number	Specification Type	Display Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Null Fill	Reserved	Right-to-Left Display	Start	End	Entry Required	Repeat	Reserved	Priority	Preprocess	1	2	3	1	2	3	Reserved	Key Mask	Reserved
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15 16	17 18	19 20	21	22	23 24	25 26	27	28	29 30	31 32	33 34	35 36	37 38	39	40	41	42	43	44	45	46	47	48 49	50 51	52 53	54 55	56 57	58 59	60 61 62 63	64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
	S																																	

Entries checked on the S-specification by WSU are:

Display Format Name (Columns 7 through 14)

This entry is the name of the display format that WSU creates from these S- and D-specifications. The *display format name* can be **1 to 8 characters** and must be left-adjusted. The first character must be alphabetic; the remaining characters must be alphameric. WSU does not allow duplicate format names within one source program.

The format name, a required entry, is the name you can specify on a PUTS operation to cause the display to appear. Refer to Chapter 7, *Coding Processing*, for an explanation of PUTS.

Format ID (Columns 15 and 16)

This 2-character field allows you to select the desired display from the WSU display and to set indicators that will allow this display to be shown only under certain conditions that you specify.

This entry can be **two alphameric characters**, other than IJ, IW, EW, EJ, or ES, that are the format ID (for example, A1, 01, or BB). Operators can select this display any time that WSU is ready for operator input. Refer to Chapter 10, *ATTENTION: OPERATORS (Running a WSU Program)*, for an explanation of how to select displays by ID. Two or more displays in one WSU program cannot have the same ID.

If the format ID entry is left **blank**, operators cannot select this display by ID.

Format ID must be blank if Reset Keyboard (columns 23 and 24) has an entry of N.

The processing level indicators (IJ, IW, EW, EJ) are optional, and you can specify only one per program. Displays assigned one of these indicators cannot be selected from the WSU display. If you want to use displays only for specific processing functions, specify:

IJ	Display occurs only when the first operator calls a WSU-generated procedure.
IW	Display occurs once for each operator (including the first operator), when that operator calls a WSU-generated procedure.
EW	Display occurs after each operator selects the end-of-work-session option on the WSU display or when the program has set on the EW indicator and completed processing for the display.
EJ	Display occurs only when the last operator selects the end-of-work-session option on the WSU display or when the program has set on the EJ indicator and completed processing for the display.

Notes:

1. Start (column 41), End (42), Repeat (44), Priority (46), Review Mode Identifying Indicators (48 through 52), and Insert Mode Record Identifying Indicators (54 through 59) must be blank if you code IJ, IW, EW, or EJ for format ID.
2. ES cannot be a format ID; however, you can code a PUTS operation in ES processing to show a display. Refer to Chapter 7, *Coding Processing*, for an explanation of the PUTS operation.

Start Line Number (Columns 17 and 18)

The number of the line at which you want this display to begin can be any number from **1 through 24** or **V**. Any data on the display that is above the starting line can neither be modified by the operator nor transmitted to your program. Single-digit entries (1 through 9) must be right-adjusted.

A **V** means that you will be using a variable starting line number. The contents of the reserved field *SLNO at the time the display is shown become the starting line number. (Refer to Chapter 3, *Reserved Fields and Indicators*, for a description of *SLNO). This entry must be left-adjusted.

For a MSG operation or IMSG operation, the contents of *SLNO at the time the display is first shown becomes the starting line number.

If any **other entry** is made, *1* is the assumed starting line number.

Use the following equation to determine the actual line number on which a field is displayed.

Actual line number	=	Start line number specified in columns 17 and 18	+	Line number specified columns 19 and 20 of field definition specification	-1

For example, if 6 is specified as the start line number and 5 is specified as the line number in the field definition (columns 19 and 20 of the D-specification), the field is actually displayed on line 10.

Note: WSU uses the bottom line (line 24) of the display for displaying messages.

Number of Lines to Clear (Columns 19 and 20)

The *number of lines to clear* (including and following the starting line specified in columns 17 and 18) before this display appears can be any number from **0 through 24**. For example, if the start line number is 6, and if 12 lines are cleared, lines 6 through 17 are cleared before the format is displayed. Single-digit entries (0 through 9) must be right-adjusted.

If column 17 contains *V* (variable starting line number), the maximum number of lines that can be cleared is 24.

If **blank or 24** is entered, the entire display is cleared.

If any **number greater than 24** is specified, a terminal error occurs. If an **invalid character** is specified, the entire display is cleared.

Lowercase (Column 21)

If a **Y** is entered, all alphabetic characters in all fields are entered in lowercase unless the operator presses the Shift key.

If a **N** is entered all characters in all fields are entered in uppercase.

If any **other** or **blank entry** is made, the entry is assumed to be N.

The D-specification entry for field lowercase always takes priority over the S-specification for field lowercase (see column 51 of the D-specification in this chapter). However, you can modify this for individual fields on the D-specification. This means that whatever is specified in the S-specification for the entire display, you can modify that definition for a particular field in the display on the D-specification.

Lowercase must be blank if Reset Keyboard (columns 23 and 24) contains an entry of N.

Return Input (Column 22)

An entry of **Y** returns all input fields to the program (including input fields in which data has not been typed) when the operator presses the Enter key, an enabled user command key, or a WSU command key.

(An exception to this rule occurs when all input fields are mandatory entry and the operator presses one of the above keys before typing data into any of the input fields. In this case, *no* input fields are returned to the program, and the current field values are retained in the WSU program. The fields are not cleared.)

An entry of **N** returns either no input fields or all input fields.

If the operator has not typed data into any of the input fields before pressing the Enter key, an enabled user command key, or a WSU command key, none of the input fields are returned to the program, and the current field values are retained in the WSU program. *The fields are not cleared.* In this situation, the operator bypasses mandatory-entry fields.

If the operator has typed data into one or more of the input fields before he presses the Enter key, a user command key, or a WSU command key, all input fields are returned to the program whether data was entered into them or not.

If a **blank** or any **other character** is entered, the entry is assumed to be N.

Return Input must be blank if Reset Keyboard (columns 23 and 24) contains an entry of N.

Note: If you specify Y for the Return Input entry, the display should have input or output/input fields.

Reset Keyboard (Columns 23 and 24)

An entry of **Y** allows processing for this display. If you do not want to allow processing for the display, enter an **N**. Displays that have a reset keyboard entry of N must not be sequenced, and their S- and D-specifications must follow the S- and D-specifications for all sequenced displays.

Refer to the PUTS operation in Chapter 7, *Coding Processing*, for an example of a display that does not allow processing.

If a **blank** or any **other character** is entered, Y is assumed and processing is allowed for this display.

Note: The reset keyboard entry must be Y if:

- Columns 15 and 16, 21 and 22, or 41 through 59 are not blank
- The display is in a sequence of displays
- The display has associated processing
- The display has input or output/input fields.

Sound Alarm (Columns 25 and 26)

An entry of **Y** causes the alarm to sound when this display appears. If you enter **N**, the alarm does not sound when this display appears.

If you would like the alarm to sound only under certain conditions, enter an indicator number **01 through 89** for the specified indicator.

If a **blank** or any **other character** or number is entered, an entry of N is assumed.

Enable Function Keys (Column 27)

You can allow the operator to use the function keys on the keyboard to control your program's operations. You use column 27 to specify whether *any* of the function keys are enabled (the operator can use them) or disabled (the operator cannot use them). The specified function keys to be enabled or disabled are identified by number in Key Mask (columns 64 through 79). The function keys are identified by number:

Function Key	Key Mask Entry
Roll Up (↑)	2
Roll Down (↓)	3

The allowed entries to enable the function keys are:

Y	The function keys identified in the key mask are enabled (allowed). If the key mask contains no entries, all function keys are disabled.
N	disable the function keys specified in the key mask. If the key mask contains no entries, the Roll ↑ (Roll Up) and Roll ↓ (Roll Down) keys are enabled.
R	The function key mask that is currently active for the display station is used when this format is displayed.
Blank	All function keys are disabled except Roll ↑ (Roll Up) and Roll ↓ (Roll Down). In this case, the key mask must not contain any entries.

Any *other* entry causes WSU to issue a warning message and assume the entry was blank.

When an operator presses an enabled function key, the WSU program either does the function or sets on the indicator that corresponds to the command key. When an operator presses a disabled function key, the system issues a message to the operator that indicates the key is not allowed at that time. No function is done for the disabled key, and the program receives no notification that the key was pressed.

Enable Command Keys (Column 28)

You can allow the operator to use the command keys on the keyboard to control your program's operations. You use column 28 to specify whether any of the command keys are enabled (the operator can use them) or disabled (the operator cannot use them). The specified command keys to be enabled or disabled are identified by a character in Key Mask (columns 64 through 79). The command keys are identified by alphabetic characters:

Command Key	Key Mask Entry
1 though 14	A through N
15 through 24	P through Y

The allowed entries to enable the command keys are:

Y The command keys identified in the key mask are enabled. If the key mask contains no alphabetic characters, all command keys are disabled.

N The command keys identified in the key mask are disabled. If the key mask entry contains no alphabetic characters, all command keys are enabled. If the operator presses a disabled command key, an error message is displayed. The operator can then press the Error Reset key, followed by the correct command key.

R The command key mask that is currently active for the display station is used when this format is displayed.

Blank All command keys are enabled. In this case, the key mask must not contain any alphabetic characters.

If any *other* entry is made, WSU issues a warning message and assumes the entry was blank.

When an operator presses an enabled command key, the WSU program either does the function (for example, shows the WSU display) or sets on the indicator that corresponds to the command key. When an operator presses a disabled command key, the system issues a message to the operator that indicates the key is not allowed at that time. No function is done for the disabled key, and the program receives no notification that the key was pressed.

Blink Cursor (Columns 29 and 30)

If you want the cursor to blink when this display appears, enter a **Y**.

If you do not want the cursor to blink when this display appears, enter an **N**.

If you want the cursor to blink only under certain conditions, you can specify an indicator number from **01 through 89**. The cursor then blinks when this display appears only if the specified indicator is on. If a **blank** or any **other entry** is made, WSU assumes the entry was an N.

Null Fill (Columns 37 and 38)

If you want blanks (hex 40s) to be changed to nulls (hex 00s) on this display, enter a **Y** or an indicator number from **01 through 89**.

If a **N** is entered, or if the entry is **blank**, null fill does not occur for this display.

Enter Mode Sequence (Columns 41 through 44)

This sequence consists of four entries:

- Start
- End
- Entry Required
- Repeat.

Note: When in review or insert mode, WSU ignores these entries.

Start (Column 41)

If this display is the first in a primary sequence or secondary sequence of displays, enter a **Y**.

If this display is not the first in a primary sequence or secondary sequence of displays, enter an **N**.

If a **blank** or any **other entry** is made, WSU assumes Y if this display is the first display (other than IJ or IW) in the program or assumes N if this display is not the first display (other than IJ) in the program.

Notes:

1. This column must be blank if columns 15 and 16 contain IJ, IW, EW, or EJ, or if Reset Keyboard (columns 23 and 24) is N.
2. A start display can also be an end display.
3. If a primary sequence *start* is not specified, WSU issues a terminal-error message.

End (Column 42)

An entry of **Y** specifies that this display is the last in a primary sequence or secondary sequence of displays.

An entry of **N** specifies that this display is not the last in a primary sequence or secondary sequence of displays.

If **blank** or any **other entry** is made, WSU assumes an entry of N.

Notes:

1. This column must be blank if columns 15 and 16 contain IJ, IW, EW, or EJ, or if Reset Keyboard (columns 23 and 24) is N.
2. An end display can also be a start display.

Entry Required (Column 43)

An entry of **Y** means that, during an enter mode display sequence, operators must press the Enter key or an enabled user command key for this display. Operators cannot use the Bypass Display command key (Cmd 2) nor can they select another display by ID from the WSU display to skip this display without causing a sequence error.

If an **N** is entered, operators can use the Bypass Display command key or select another display by ID from the WSU display to skip this display.

If any **other entry** is made, WSU assumes an entry of N.

Repeat (Column 44)

An entry of **Y** causes this display to repeat during an enter mode display sequence and to accept input until:

- An operator presses the Bypass Display command key (Cmd 2).
- An operator presses the WSU Display command key (Cmd 1) and selects a display by typing a format ID.
- A PUTS operation causes another display to appear.

You can repeat any display: a display in a primary sequence, a display in a secondary sequence, or a nonsequenced display.

An entry of **N** causes this display to not repeat.

If **blank** or any **other entry** is made, WSU assumes an entry of N.

Note: This entry must be blank if columns 15 and 16 contain IJ, IW, EW, or EJ.

Priority (Column 46)

If you want to specify the relative expected frequency of this display, you can enter a number from **0 through 3**. 3 indicates a display that you expect to use most frequently; 2 or 1 indicates a display that you expect to use less frequently, and 0 indicates a display that you expect to use least frequently.

If any **other entry** is made, WSU assumes an entry of 0.

Note: This entry must be blank if columns 15 and 16 contain IJ, IW, EW, or EJ.

Preprocess (Column 47)

An entry of **Y** causes WSU to begin performing the C-specifications for this display before showing the display. A preprocessed display must have associated C-specifications.

If an **N**, **blank**, or any **other entry** is made, preprocessing does not occur for this display. Refer to the explanation of the PUTS operation in Chapter 7, *Coding Processing*, for examples of a preprocessed display.

Review Mode Record Identifying Indicators

(Columns 48 through 53)

If you specify the type of record in the transaction file that the operators can review with this display, you can enter any indicator number from **01 through 89**. This entry should match the record identifying indicator on the I-specification for the type of record to review.

You can specify as many as three types of records that can be reviewed with this display.

Refer to Chapter 2, *How WSU Works*, for an explanation of review mode.

If columns 48 through 53 are **blank**, operators cannot select this display in review mode. This display can, however, be shown in review mode by means of a PUTS operation.

Note: This entry must be blank if columns 15 and 16 contain IJ, IW, EW, or EJ.

Insert Mode Record Identifying Indicators

(Columns 54 through 59)

To specify the type of record in the transaction file after which operators can use this display to insert a record, you can enter an indicator number from **01 through 89**. This entry should match the record identifying indicator entry on the I-specification for that type of record.

These insert mode indicators must also be used as review mode indicators (columns 48 through 53) in the program.

You can specify as many as three types of records on one S-specification.

If columns 54 through 59 are **blank**, operators cannot select this display in insert mode. This display can, however, be shown in insert mode by means of a PUTS operation.

Refer to Chapter 2, *How WSU Works*, for an explanation of insert mode.

Note: This entry must be blank if columns 15 and 16 contain IJ, IW, EW, or EJ.

Key Mask (Columns 64 through 79)

The key mask represents the command and function keys that are enabled or disabled when this display appears. The mask is a string of 1 to 16 letters and numbers from the following list:

Mask ID	**Command/Function Key Symbol**	**Description**
A	Cmd 1	WSU Display command key
B	Cmd 2	Bypass Display command key
C	Cmd 3	Resume Entry Mode command key
D	Cmd 4	Insert Mode command key
E	Cmd 5	Page Backward Group command key
F	Cmd 6	Page Forward Group command key
G	Cmd 7	User command key 7 (Indicator KG)
H	Cmd 8	User command key 8 (Indicator KH)
I	Cmd 9	User command key 9 (Indicator KI)
J	Cmd 10	User command key 10 (Indicator KJ)
K	Cmd 11	User command key 11 (Indicator KK)
L	Cmd 12	User command key 12 (Indicator KL)
M	Cmd 13	Accept-Sequence-Error command key
N	Cmd 14	Delete Mode command key
P	Cmd 15	Resume Review Mode command key
Q	Cmd 16	User command key 16 (Indicator KQ)
R	Cmd 17	User command key 17 (Indicator KR)
S	Cmd 18	User command key 18 (Indicator KS)
T	Cmd 19	User command key 19 (Indicator KT)
U	Cmd 20	User command key 20 (Indicator KU)
V	Cmd 21	User command key 21 (Indicator KV)
W	Cmd 22	User command key 22 (Indicator KW)
X	Cmd 23	User command key 23 (Indicator KX)
Y	Cmd 24	User command key 24 (Indicator KY)
2	Roll ↑	Page Forward Record function key
3	Roll ↓	Page Backward Record function key

In this list, A through F and M through P correspond to WSU command keys; G through L and Q through Y correspond to user command keys (Indicators KG through KL and KQ through KY); and 2 and 3 correspond to the Page Forward Record and Page Backward Record function keys.

The mask must be left-adjusted and must not have embedded blanks. The letters and numbers in the mask can be in any order; however, duplicate entries cause a warning message to be printed.

If the mask contains embedded blanks, a warning message is issued, and WSU ignores all entries that follow the first blank.

Refer to Chapter 10, *ATTENTION: OPERATORS (Running a WSU Program)*, for an illustration of the command/function keys.

The entry for Enable Function Keys in column 27 specifies whether the function keys in the mask are enabled or disabled when this display appears. The Enable Command Keys entry in column 28 specifies whether the command keys in the mask are enabled or disabled when this display appears.

If you code a digit in the mask, column 27 must not be blank or R. If you code a letter in the mask, column 28 must not be blank or R. If columns 27 and 28 are both blank, columns 64 through 79 are ignored.

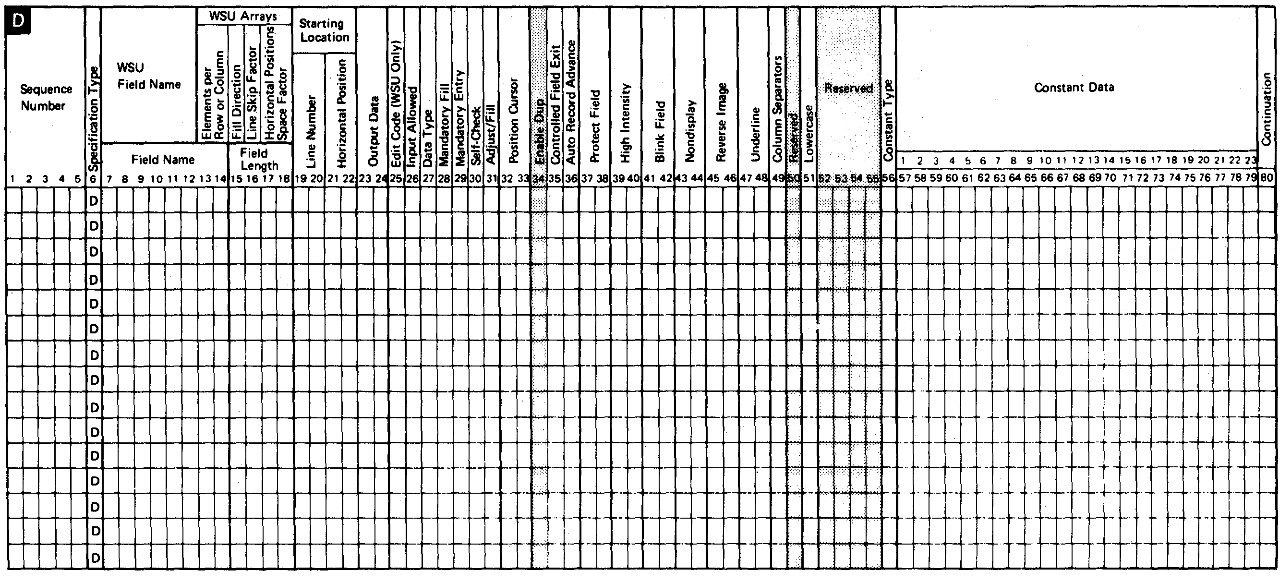
If any WSU command keys (A through F and M through P) or the Roll ↑ (Roll Up) and Roll ↓ (Roll Down) function keys are disabled or not enabled, WSU issues a warning message. If any of these keys are disabled, problems may occur during program operation. For example, if you disable the WSU Display command key, an operator cannot use that key to display the WSU display. Therefore, he cannot end his session, and the WSU program has to end the session by setting on the EW or EJ indicator.

D-Specification

A summary of the entries allowed on the D-(field definition) specification is covered in the remainder of this chapter. The D-specifications are coded on the lower section of the Display Format specification sheet. Columns 50 and 52 through 55 are reserved; entries in these columns are ignored by WSU. If any entry is made in column 34, WSU issues a warning message and continues as though it was blank.

When entering the WSU specifications on the System/36, the D-specification must be entered after the S-specification and before the C-specifications (Chapter 17).

Field Definition Specification



WSU uses the following entries on the D-specification:

WSU Field Name (Columns 7 through 12)

The **WSU field name** is the name of an input field, output field, or output/input field, or an array name. The field name can be **1 to 6 characters** and must be left-adjusted. The first character must be alphabetic or &; the remaining characters must be alphameric. Embedded blanks are not allowed. This entry can be a field name from an I-specification; a result field name on a C-specification; an array name; or UDATE, UYEAR, UDAY, or UMONTH.

If this column is **blank**, this D-specification statement defines only constant data.

Notes:

1. If the field name begins with an &, the character following the & must be alphabetic.
2. You can code the same field name on more than one D-specification line; however, you cannot overlap data on the display. Refer to the description of Horizontal Position (columns 21 and 22) in this chapter for a description of overlapping data.

3. A field name that begins with * should not be coded on the D-specification because WSU considers a line beginning with an * to be a comment.

WSU Arrays (Columns 13 through 18)

WSU execution-time arrays use columns 13 through 18 of the D-specification. If you are not defining arrays, leave columns 13 through 18 blank. Refer to Chapter 5, *Coding Arrays*, and Chapter 14, *E-(Array Specifications)*, for more information about using execution-time arrays in WSU programs.

Array Elements per Row or Column (Columns 13 and 14)

Any number of array elements from **01 through 99** per row or column can be entered.

Fill Direction (Column 15)

An entry of **R** in position 15 references the rows of the array. An entry of **C** references the columns of the array.

The R or C coded with an array name references the first element of the array and specifies that the D-specifications are to be generated to format rows or columns on the display for output to and/or input from individual array elements.

Line Number Skip Factor (Column 16)

Any number from **0 through 99** can be entered to specify the number of rows between elements of the array.

Horizontal Positions Space Factor (Columns 17 and 18)

Any number from **00 through 99** can be entered to specify the number of columns between elements of the array.

Starting Location (Columns 19 through 22)

The starting location for a WSU field consists of:

- A line number (columns 19 and 20)
- A horizontal position (columns 21 and 22).

Line Number (Columns 19 and 20)

The line number can be a number from **1 through 25**, and is the line on which the data begins on the display. WSU calculates the actual line number for the data by adding the display's starting line number to this line number minus one.

The maximum entry is 25 minus the display's starting line number. During a program, when a display that has a variable starting line number is shown, the sum of the starting line number and the number of the last line used for that display cannot exceed 24. A terminal error appears if this situation occurs.

WSU issues a warning message during program generation if data is coded on an uncleared portion of the display.

If the line number is 1, the horizontal position cannot be 1.

A line number is a required entry. Single-digit entries must be right-adjusted.

Horizontal Position (Columns 21 and 22)

The position of the data on a line can be any number from **1 through 80**. If this D-specification statement specifies a prompt and a field, this entry specifies the beginning position of the prompt. One blank is inserted to separate the last position of the prompt and the first position of the field. Single-digit entries must be right-adjusted.

Notes:

1. You cannot use position 01 of line 01.
2. Data cannot overlap (occupy the same position) on the display. You must not code the same line number and horizontal position for two or more fields or prompts, or a line number and horizontal position that is within another field or prompt.
3. At least one blank must separate fields or a field and prompt.

Output Data (Columns 23 and 24)

A **Y** specifies that data from the field named in Field Name (columns 7 through 12) is to be displayed.

An **N** specifies that the data from the field named in columns 7 through 12 is not to be displayed.

An entry of an indicator number from **01 through 89** causes the data from the field named in columns 7 through 12 to be displayed only if the specified indicator is on. If the D-specification indicates a prompt in positions 57 through 79 for the field named in positions 7 through 12, the prompt is displayed whether the indicator is on or off.

If a **blank** or an **other entry** is made, WSU assumes an entry of N.

Note: This entry applies only to a field named in columns 7 through 12. Therefore, this entry should be blank or N if column 56 is M, C, or D.

Edit Code (Column 25)

An entry of **J** inserts commas, decimal points, and a minus sign (for a negative field) in the numeric output field.

An entry of **Y** omits leading zeros of a numeric output field and inserts slashes (/) for each pair of digits from left to right (if column 43 of the J-specification is M or D) or from right to left (if column 43 of the J-specification is Y). Refer to *Date/Edit (Column 43)* in Chapter 13.

This entry applies to numeric output fields that are 3 to 15 digits long.

If you specify a **Z**, leading zeros from the numeric output field are suppressed. An all-zero field is displayed with all blanks.

If a **blank** or any **other entry** is made, editing of the output field is not allowed. This entry must be blank if the WSU Field Name (columns 7 through 12) is blank.

Note: Edit codes are allowed only for numeric output fields. An edited output field cannot be an input field. Input Allowed (column 26) must not be Y.

Input Allowed (Column 26)

An entry of **Y** specifies that the field named in columns 7 through 12 is an input field.

An entry of **N** specifies that this D-specification statement does not describe an input field. Columns 27 through 38 must be blank.

If a **blank** or any **other entry** is made, WSU assumes an entry of N.

Data Type (Column 27)

An entry in column 27 can specify the following data types:

A Only alphabetic data can be entered in the input field.

B Only alphameric data can be entered in the input field.

D Input data typed in this field can contain only the digits 0 through 9 and overscored asterisks ($\overline{*}$).

E This field can contain alphameric and Katakana characters (A/N/K), or ideographic characters (IGC), but not both. The field is initially set to binary zeros and the display station is set for alphameric data entry. When the cursor is in the first position of the field it blinks. The blinking indicates that the operator can change modes and enter ideographic data.

F This field can contain alphameric, Katakana, or ideographic data, but only one at a time. The field is initially filled with ideographic nulls (a shift-out character, blanks, and a shift-in character), and the display station is set for ideographic entry.

K This field can contain Katakana characters.

N Numbers 0 through 9, plus and minus signs, commas, periods, and blanks can be entered in this input field.

O Operators can enter a combination of alphameric, Katakana, and ideographic data. The field is initially set to alphameric and filled with binary zeros.

S Numbers 0 through 9 can be entered in this input field.

A field exit key must be pressed after entering the field, regardless of the entry for Controlled Field Exit (column 35).

Pressing the Field+ or Field Exit key enters a field with a plus sign in the zone portion of the rightmost digit; pressing the Field- key enters a field with a minus sign in the zone portion of the rightmost digit. The sign, therefore, is not entered and does not occupy an extra position in the field. When a field exit key is pressed, the data is right-adjusted, and unused positions appear as blanks and transmit as zeros regardless of the entry for Adjust/Fill (column 31).

If you display a negative signed numeric field with a J edit code (column 25), a minus sign occupies a separate position on the display at the end of the number (for example 123-).

If you display this field without a J edit code, the rightmost digit contains the sign. (For example, 5- is displayed as the character N. The zone portion contains the minus sign and the digit contains 5.)

If you display a positive signed numeric field with a J edit code, a blank follows the rightmost digit. If you display this field without a J edit code, the last digit is not followed by a blank.

X The field can contain only ideographic data.

Code X data type fields as input fields. On the D-specification, enter an N in column 23, or leave columns 23 or 24 blank.

If a **blank** or any **other entry** is made, WSU assumes B for an alphameric field and S for a numeric field (you must allow an extra position on the display for a sign).

For data types E, F, K, and O, you must have coded a Y in column 44 of the J specification.

This data-type entry overrides the data type specified on the C-specification or the I-specification, if the data types are not the same. If you specify constant data in columns 57 through 79, ensure that data is the type you specify in the data-type entry.

When processing numeric fields, your program might produce unpredictable results for fields that have data types other than S. These data types can allow input that is not numeric, such as blanks, commas, and periods. Your program is responsible for checking that only numeric data is entered into those numeric fields.

Mandatory Fill (Column 28)

When mandatory fill is specified, if one character is typed in an input field, all positions in the field must be filled with non-null characters. (A non-null character is any character that can be entered from the keyboard or a space produced by the operator pressing the space bar.) The operator can use the cursor keys to exit from a mandatory fill field. However, if the operator uses the cursor keys, no adjusting will occur.

An entry of **Y** specifies that if one character is typed in the field, all positions in the field must be filled with non-null characters. If you specify mandatory fill, you cannot specify adjust/fill (column 31 of the D-specification) for the same field.

An entry of **N** or **blank** specifies that this input field does not have to be filled.

If any **other entry** is made, WSU will assume an entry of N.

Mandatory Entry (Column 29)

An entry of **Y** specifies that at least one character or blank must be entered in the input field.

An entry of **N** specifies that this input field can be bypassed on this display.

If a **blank** or any **other entry** is made, WSU assumes an entry of N.

Notes:

1. This entry applies during enter, review, and insert modes.
2. A mandatory-entry field that is blank when a display is entered causes a keyboard error to appear. The operator must press the Error Reset key and then press the Field Backspace key to return the cursor to the beginning of the mandatory-entry field. An operator can bypass a mandatory-entry field if:
   - All input fields on the display are mandatory-entry fields, the Return Input entry on the S-specification is Y, and the operator does not enter data in any of the input fields.
   - The Return Input entry on the S-specification is N and the operator does not enter data in any of the input fields.

Self-Check (Column 30)

A **T** specifies that the input field is a modulus 10 self-check field.

An **E** specifies that the input field is a modulus 11 self-check field.

If a **blank** or any **other entry** is made, WSU assumes that the input field is not a self-check field. This entry must be blank for alphameric fields.

When a check digit error occurs, a 4-digit message number flashes on line 24. When the operator presses the Error Reset key, the cursor returns to the start of the field in error (the field is not changed), and the operator can enter the data again.

Refer to *Appendix A. Using Self-Check Fields* for an explanation of self-check fields.

Adjust/Fill (Column 31)

An entry of **Z** causes the field to be right-adjusted and filled with zeros. WSU right-adjusts data that operators enter into the field; unused positions appear and transmit as zeros.

An entry of **B** causes the field to be right-adjusted and filled with blanks. WSU right-adjusts data that operators enter into the field; unused positions appear and transmit as blanks.

If the field is signed-numeric, unused positions appear as blanks but transmit as zeros.

An entry of either Z or B causes the field to have controlled field exit regardless of the entry in column 35 of this D-specification. The operator must press one of the field exit keys after entering the data in order to enter the field.

If a **blank** or any **other entry** is made, WSU assumes B for a signed numeric field. Adjust/fill is not done for other fields.

Note: Operators can press the Field+ key (for numeric or signed numeric fields), the Field- key (for signed numeric fields), or the Field Exit key (for adjust/fill fields). Operators can press the Field Advance key for an adjust/fill field, but the adjust/fill does not occur.

Position Cursor (Columns 32 and 33)

If a **Y** is entered, the cursor is positioned at the beginning of the input field when this display appears. If you code Y for more than one field, the cursor positions itself at the beginning of the first field on the D-specification for which you coded Y.

If an **N** is entered, the cursor is not positioned at the beginning of the input field when this display appears.

An indicator number entry of **01 through 89** positions the cursor at the beginning of the input field when this display appears only if the specified indicator is on.

You can use indicators to position the cursor for all, some, or none of the fields on a D-specification.

If two or more indicators that position the cursor are on, the cursor is at the beginning of the field on the first D-specification for which you specified one of those indicators.

If **blank** or any **other entry** is made, WSU assumes an entry of N.

Note: If you do not position the cursor at any field on the display, the cursor appears at the beginning of the first unprotected field on the D-specification, or the cursor remains in its current position if the display contains no unprotected fields.

Controlled Field Exit (Column 35)

An entry of **Y** specifies that one of the field exit keys (Field Adv, Enter, Field Exit, Field+, Field- [if the field is a signed-numeric field], Field Backspace, Home, or Erase Input) must be pressed before the cursor will leave the field.

An entry of **N** specifies that the cursor automatically exits from the field when the field is filled, unless adjust/fill is specified for the field in column 31 or an S data type is specified in column 27.

If a **blank** or any **other entry** is made, WSU assumes an entry of Y if adjust/fill or an S data type is specified for the field. Or it assumes an entry of N if adjust/fill or an S data type is not specified for the field.

Note: An entry of N does not cause an automatic skip from the last field of this display to the first field of the next display. Instead, the cursor returns to the first unprotected field on this display. The entry in Auto Record Advance in column 36 can cause an automatic advance to the next display.

Auto Record Advance (Column 36)

An entry of **Y** specifies that all input fields on this display are automatically entered when:

- Operators enter the last character of the last input field, or
- The cursor is in the last input field and operators press the Field Exit, Field+, or Field- key (for a signed numeric field).

An entry of **N** specifies that automatic-record-advance does not occur for this field.

If a **blank** or any **other entry** is made, WSU assumes an entry of N.

Protect Field (Columns 37 and 38)

An entry of **Y** specifies that the cursor skips the field. Data cannot be entered in protected fields. You can protect any field on a display.

An entry of **N** specifies that this field is not protected, and the operator can type data in this field.

If an indicator number from **01 through 89** is entered, the cursor skips the field only if the specified indicator is on.

If a **blank** or any **other entry** is made, WSU assumes an entry of N for an input field; it assumes an entry Y for an output-only field.

Notes:

1. WSU automatically protects output-only fields.
2. If you protect the following fields, WSU sets them to blank or zero when the display is entered:
   - Input-only fields
   - Output/input fields that have output conditioned by an indicator that was off when the display appeared.

   You may unintentionally lose data by protecting these fields.
3. The cursor may appear in a protected field if all of the following conditions exist:
   - The field is protected by an indicator that is on when the display appears.
   - The field is the first field defined on the D-specifications.
   - The cursor is not put in position by means of an indicator to any other field (columns 32 and 33 for all other fields are blank or contain indicators that are off). To avoid putting the cursor in this position, use the same indicator to protect this field and to position the cursor in the next field on the display.

4. When a display is shown again by means of an MSG operation, fields protected on the display remain protected; fields not protected on the display remain unprotected.

5. If a field is defined as either or both nondisplay and protected (Y in columns 43 and 37 of the D-specification) and if column separators are requested (Y in column 49 of the D-specification), the column separators are displayed on a 5251 Display Station and on a 5291 Display Station; the column separators are **not** displayed on a 5292 Color Display Station.

6. If an indicator is used to control both the nondisplay and protect attributes and if column separators are requested, the separators are not displayed if both indicators are on when the field is displayed.

High Intensity (Columns 39 and 40)

An entry of **Y** causes the data to be intensified (made brighter than the normal intensity).

An entry of **N** specifies that the data is not to be intensified.

If an indicator number from **01 through 89** is specified, the data is intensified only if the specified indicator is on.

If a **blank** or any **other entry** is made, the data is not intensified.

If this display format is shown on a 5292 Color Display Station and high intensity is specified, the field is displayed with white characters. If other field attributes are specified, the color result is different. For the color result of specific attribute combinations, see Figure 16-1. For additional information about the control of color, see the *5292 Color Display Station Programmer's Guide*.

WSU allows high intensity to be conditioned by different indicators. If all indicators are on when the display appears, the field is not displayed. Refer to Figure 16-2 for a list of invalid combinations of attributes.

Blink Field (Columns 41 and 42)

An entry of **Y** specifies that the field blinks when the display appears.

An entry of **N** specifies that the field does not blink when the display appears.

An entry of an indicator number from **01 through 89** specifies that this field blinks only if the specified indicator is on.

If a **blank** or any **other entry** is made, WSU assumes an entry of N, and the field does not blink.

If this format is displayed on a 5292 Color Display Station and blink field is specified, the field is displayed with red characters, and does not blink. To cause the characters in the field to blink, you must also specify high intensity (columns 39 and 40 of the D-specification). If other field attributes are also specified, the color result might be different. For the result of specific attribute combinations, see Figure 16-1. For additional information about the control of color, see the *5292 Color Display Station Programmer's Guide*.

Notes:

1. If nondisplay is specified and high intensity (columns 39 and 40), reverse image (columns 45 and 46), or underline (columns 47 and 48) is also specified, the field is defined as nondisplay only.
2. If a field is defined as either or both nondisplay and protected (Y in columns 43 and 37 of the D-specification) and if column separators are requested (Y in column 49 of the D-specification), the column separators are displayed on a 5251 Display Station and on a 5291 Display Station; the column separators are **not** displayed on a 5292 Color Display Station.
3. If an indicator is used to control both the nondisplay and protect attributes and if column separators are requested, the separators are not displayed if both indicators are on when the field is displayed.

Refer to Figure 16-2 for a list of invalid combinations of attributes.

Nondisplay (Columns 43 and 44)

An entry of **Y** specifies that the data from the field is not displayed when the display appears or that the data entered into the field is not displayed.

An entry of **N** specifies that the data be displayed.

An indicator number from **01 through 89** specifies that this data is displayed only if the specified indicator is on.

If **blank** or any **other entry** is made, WSU assumes an entry of N and the data is displayed.

See Figure 16-2 for a list of invalid combinations of attributes.

Color Result	Attributes Specified				Underline can be Specified
	Column Separators[2]	Blink Field	High Intensity	Reverse Image	
Green					X
Green, reverse image				X	X
White			X		X
White, reverse image			X	X	
Red		X[1]			X
Red, reverse image		X[1]		X	
Red, blink		X	X		X
Red, reverse image, blink		X	X	X	
Turquoise, column separators	X				
Turquoise, reverse image, column separators	X			X	X
Pink	X[3]	X[1]			X
Pink, reverse image	X[3]	X[1]		X	X
Yellow, column separators	X		X		X
Yellow, reverse image, column separators	X		X	X	
Blue	X[3]	X[1]	X		X
Blue, reverse image	X[3]	X[1]	X	X	
Data in fields with these			X	X	X
combinations of attributes		X	X	X	X
are not displayed.	X		X	X	X
	X	X	X	X	X

[1] Field does not blink.

[2] Column separators do not appear when reduced line spacing is used.

[3] Column separators do not appear.

Figure 16-1. Controlling Color on a 5292 Color Display Station

Notes:

1. Underlines and column separators are always blue.
2. Underlines do not blink if blink field is also specified.
3. Column separators do not appear if blink field is also specified.
4. Use the limited color switch of the 5292 Color Display Station to see how a display format designed for color will appear on a single-color display.

Reverse Image (Columns 45 and 46)

An entry of **Y** specifies that the data image be reversed (characters are dark on a light background).

An entry of **N** specifies that the image not be reversed (characters are light on a dark background).

An indicator number entry from (b01 through 89) b specifies that the data image be reversed only if the specified indicator is on.

If a **blank** or any **other entry** is made, WSU assumes an entry of N and the image is not reversed.

WSU allows reverse image to be conditioned by different indicators. If all indicators are on when the display appears, the field is not displayed. Refer to Figure 16-2 for a list of invalid combinations of attributes.

Attributes	**WSU Action**
Mandatory fill and adjust/fill	Mandatory fill only
High intensity and nondisplay	Nondisplay only
Blink field and nondisplay	Nondisplay only
Reverse image and nondisplay	Nondisplay only
Underline and nondisplay	Nondisplay only
Reverse image, underline, and high intensity	Terminal error-message
Mandatory entry and protect field	Protect field only

Figure 16-2. Invalid Combinations of Attributes

Underline (Columns 47 and 48)

An entry of **Y** specifies that the data on the display be underlined.

An entry of **N** specifies that the data not be underlined.

An indicator number entry of **01 through 89** specifies that this display be underlined only if the specified indicator is on.

If a **blank** or any **other entry** is made, WSU assumes an entry of N and the data is not underlined.

If this format is displayed on a 5292 Color Display Station and underline is specified, the field is displayed with a blue line beneath the character positions in the field. The color of the characters displayed in the field depends on the other field attributes that are specified. For the result of specific attribute combinations, see Figure 16-1. For additional information about the control of color, see the *5292 Color Display Station Programmer's Guide.*

WSU allows underline to be conditioned by different indicators. If all indicators are on when the display appears, the field is not displayed. Refer to Figure 16-2 for a list of invalid combinations of attributes.

Column Separators (Column 49)

An entry of **Y** specifies that a divider be placed on the display before and after each character position in the field. The vertical lines do not occupy a character position.

For example:

Before operator data entry: .‾. . .
After operator data entry: .1.3.5.

An entry of **N** specifies that column separators not be used.

If a **blank** or any **other entry** is made, WSU assumes an entry of N and the column separators are not used.

If this format is displayed on a 5251 Display Station, the column separators appear as vertical lines (|) on either side of each character position in the field.

If this format is displayed on a 5291 Display Station, the column separators appear as two vertical dots (:) on either side of each character position in the field.

If this format is displayed on a 5292 Color Display Station, the column separators appear as blue dots at the bottom corners of each character position in the field. If blink field is also specified, the column separators do not appear on the display. The color of the characters displayed in the field depends on the other field attributes that are also specified. For the result of specific attribute combinations, see Figure 16-1. For additional information about the control of color, see the *5292 Color Display Station Programmer's Guide.*

Notes:

1. If a field is defined as either or both nondisplay and protected (Y in columns 43 and 37 of the D-specification) and if column separators are requested (Y in column 49), the column separators are displayed on a 5251 Display Station and on a 5291 Display Station; the column separators are not displayed on a 5292 Color Display Station.

2. If an indicator is used to control both the nondisplay and protect attributes and if column separators are requested, the separators are not displayed if both indicators are on when the field is displayed.

Lowercase (Column 51)

An entry of **Y** specifies that all alphabetic characters typed into this field from the keyboard can be displayed and sent to the user program in lowercase (if the Shift key is not pressed when the character is typed in by the operator) or in uppercase (if the Shift key is pressed when the character is typed in by the operator).

An entry of **N** specifies that all characters typed from the keyboard be sent to the program in uppercase.

Lowercase must be blank for numeric fields.

The D-specification entry for field lowercase always takes priority over the S-specification for field lowercase (see column 21 of the S-specification in this chapter). This means that whatever is specified in the S-specification for the entire display, you can modify that definition for a particular field in the display.

The following chart defines what can be typed in the input field depending on what is specified in the S- and D-specifications:

Entry in Column 51 of the D-Specification	**Entry in Column 51 of the S-Specification**		
	Blank	**N**	**Y**
Blank	Uppercase only	Uppercase only	Lowercase or uppercase
N	Uppercase only	Uppercase only	Uppercase only
Y	Lowercase or uppercase	Lowercase or uppercase	Lowercase or uppercase

Constant Type and Constant Data (Columns 56 through,79)

Columns 56 through 79 are for specifying the constants (or prompts) used on your display formats. Refer to Chapter 6, *Coding Displays*, for more about using constants. Figure 6-7 is a D-specification coding chart. Constant types C, D, and P can use ideographic characters while constant types F and M cannot.

If column 56 contains a blank or F, columns 57 through 79 must be blank.

C An entry of **C** in column 56 indicates that columns 57 through 79 contain an initial constant value for the input field named in columns 7 through 12. Other required entries are:

- The input field name in columns 7 through 12.
- The starting location of the constant in columns 19 through 22.
- Output Data in columns 23 and 24 must be N or blank.
- A Y in Input Allowed in column 26.
- The constant in columns 57 through 79. This constant, which must *not* be enclosed with apostrophes, can be continued to columns 7 through 79 of the following D-specification statement.

WSU displays the constant when the display appears. If a data type is specified in column 27, WSU does not verify that the constant is of the same type. An alphameric field that has a constant initial value is replaced with the value, beginning with the leftmost position. Positions that are not entered remain unchanged.

D An entry of *D* in column 56 indicates that columns 57 through 79 contain an initial constant value for the input field named in columns 7 through 12, *and* the field's name is used as the prompt for the field. Required entries are:

- The input field name in columns 7 through 12.
- The starting location of the field name prompt in columns 19 through 22.

  Regardless of the length of the field name, WSU reserves 6 positions on the display for the name. In addition, one blank separates the last position for the field name from the first position of the constant initial value.
- Output Data in columns 23 and 24 must be N or blank.
- A Y in Input Allowed (column 26).
- The constant in columns 57 through 79. This constant, which must *not* be enclosed with apostrophes, can be continued to columns 7 through 79 of the following D-specification statement.

WSU displays the constant and the field name when the display appears. The cursor skips over the field name during data entry and positions itself at the beginning of the constant.

If a data type is specified in column 27, WSU does not verify that the constant is of the same type. An alphameric field that has a constant initial value is replaced with the value, beginning with the leftmost position. Positions that are not entered remain unchanged.

P An entry of *P* in column 56 indicates that columns 57 through 79 contain a prompt that starts in the position specified in columns 19 through 22. Required entries are:

- The starting location of the prompt in columns 19 through 22.
- The prompt, enclosed in apostrophes, in columns 57 through 79. This prompt can be continued to columns 7 through 79 of the following D-specification statement by coding a nonblank character in position 80.

A field name may be coded on the same D-specification as a prompt. The starting location refers to the prompt. One blank separates the last position of the prompt and the first position of the field.

The cursor skips over each prompt on a display during data entry and positions itself only at the beginning of input fields.

F An entry of *F* in column 56 specifies that the field name in columns 7 through 12 becomes the prompt for the field. Required entries are:

- The name of an input field, output field, or output/input field in columns 7 through 12.
- The starting location of the field-name prompt in columns 19 through 22.

  Regardless of the length of the field name, WSU reserves 6 positions on the display for the name. In addition, one blank separates the last position for the field name from the first position of the field.

The cursor skips over each prompt on a display during data entry and positions itself only at the beginning of input fields.

M An entry of *M* in column 56 indicates that columns 57 through 60 contain a 4-digit message identification code (MIC) that references a message in your message load member. Required entries are:

- Field Name in columns 7 through 12 must be blank.
- The starting location of the prompt in columns 19 through 22.
- Output Data in columns 23 and 24 must be blank.
- The 4-digit MIC in columns 57 through 60.

Columns 23 through 30 of the J-specification specify the message load member to use.

WSU determines the number of positions between the start of this message and the start of the next field or constant on this display. If you do not leave enough room for your message, the extra rightmost characters are truncated. You must allow at least 6 positions for the message on the display.

The cursor skips over each message on a display during data entry and positions itself only at the beginning of input fields.

Continuation (Column 80)

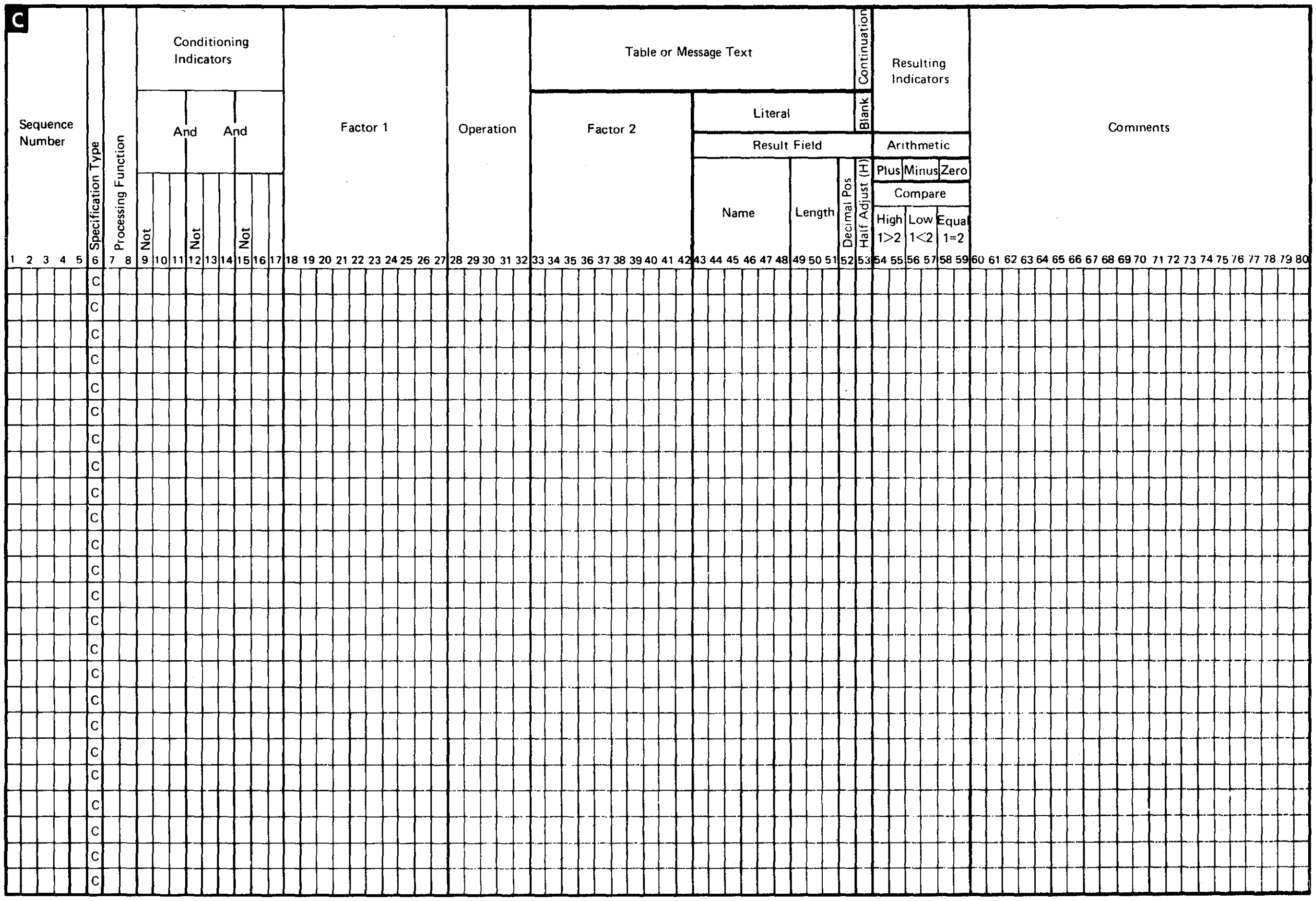
This entry can be any character except a blank; it allows you to continue coding constant data that does not fit in columns 57 through 79 into columns 7 through 79 of the following D-specification statement. You are allowed one continuation of any D-specification statement that has C, P, or D in column 56. The continuation line should not contain * in column 7, because WSU then assumes that the line contains a comment rather than a continuation of the previous line.

Chapter 17. C-(Processing) Specification

This chapter summarizes the entries that you can code on the C-specification. You can find additional information about coding C-specifications in the following:

- Chapter 2 describes the WSU program cycle and processing levels.
- Chapter 3 explains the indicators and reserved fields you can use on the C-specification.
- Chapter 7 describes coding processing.
- Chapter 11 contains example programs for which C-specifications have been coded.
- Chapter 18 contains detailed explanations of the operation codes.

S/36 WSU Processing Specifications



WSU checks the following entries on the C-specification.

Processing Function (Columns 7 and 8)

Below are the processing functions that you can specify for your program.

Blank	Operation for the display defined by the S- and D-specifications that come before these C-specifications, or operation in a subroutine.
J	Operation that occurs only during the job-initiation processing level.
W	Operation that occurs only during the work-session-initiation processing level.
ES	Operation that occurs only during the end-of-sequence-set processing level. ES processing occurs only for the primary display sequence.
EW	Operation that occurs only during the end-of-work-session processing level.
EJ	Operation that occurs only during the end-of-job processing level.
SR	Operation in a subroutine. SR in columns 7 and 8 is an optional entry for a subroutine operation, including BEGSR, PRTY, and ENDSR.
AN or OR	AND/OR relationship line used to link conditioning indicators in columns 9 through 17. Use the AN and OR entries to group as many as seven OR lines, seven AND lines, or seven lines with any combination of the two to condition an operation. The first line of the group can have blanks, IJ, IW, ES, EW, EJ, or SR in columns 7 and 8. All lines after the first must have AN or OR in columns 7 and 8. The last line of the group contains the operation and the necessary operands. All lines in the group except the last line must have blanks in columns 18 through 59, and must have one or more indicators in columns 9 through 17.

Note: If you code a processing-function indicator in columns 15 and 16 of an S-specification, all C-specifications associated with that S-specification must have the same processing function specified in columns 7 and 8.

Conditioning Indicators (Columns 9 through 17)

The following indicators can be used to specify the conditions for doing the operations you want done for your program.

Indicator	**Operation is done:**
Blank	Each time the C-specification is processed.
AC	When one of the KG through KL and KQ through KY command key indicators is on.
AE	When the AE (accept sequence error) indicator is on.
CG	When the CG (current group) indicator is on.
DL	When the DL (delete) indicator is on.
EJ	When the EJ (end of job) indicator is on. This operation can occur outside of the EJ processing level if the EJ indicator has been set on by means of a SETON operation.
ES	When the ES (end of sequence set) indicator is on. This operation can occur outside of the ES processing level if the ES indicator has been set on by means of a SETON operation.
EW	When the EW (end of work session) indicator is on. This operation can occur outside of the EW processing level if the EW indicator has been set on by means of a SETON operation.
IJ	When the IJ (job initiation) indicator is on.
IP	When the IP (input to process) indicator is on.
IN	When the IN (insert mode) indicator is on.
IS	When the IS (start transaction sequence) indicator is on.
IW	When the IW (work session initiation) indicator is on.
JA-JN, JP-JY	When the specified job indicator is on.
KG-KL, KQ-KY	If the corresponding command key was enabled and pressed.
PG	When the PG (program mode) indicator is on.
RC	When an operator restarts a work session after a system or program error. The program must have a transaction file in order to use this indicator.
RP	For a display that has been displayed again by means of the MSG operation.
RR	When the RR (record review) indicator is on.

RS	When an operator restarts a work session after signing off. The program must have a transaction file in order to use this indicator.
RU	When the RU (rover/update) indicator is on.
RV	Only in review mode or insert mode.
SA-SN, SP-SY	When the specified session-level indicator is on.
U1-U8	If the specified indicator was set on by means of the SWITCH OCL statement before the program ran or by means of a SETON operation.
01-89	If the specified indicator is on.

Use columns 9 through 17 to assign indicators that specify the conditions for doing the operation. You can code from one to three indicators in columns 10 and 11, 13 and 14, and 16 and 17 on each line. If an indicator must be *off* before the operation is done, code an N before the indicator in column 9, 12, or 15.

The indicators in columns 9 through 17 operate in an *and* relationship with one another. By coding an AN or OR entry in columns 7 and 8, you can use more than three indicators to condition a single operation. You can include as many as seven OR lines, seven AND lines, or seven lines with any combination of the two.

For more information about indicators, refer to Chapter 3, *Reserved Fields and Indicators*.

Factor 1 and Factor 2 (Columns 18 through 27 and 33 through 42)

These entries name fields or array elements or specify data (literals) for an operation in columns 28 through 32. Allowed entries depend upon the operation. The maximum length of a file/format name is 8; the maximum length for a field/label name is 6. Figure 17-1 shows the possible entries for factor 1 and factor 2 for each operation.

Operation Type	Operation Code (28 through 32)	Factor 1 (18 through 27)	Factor 2 (33 through 42)
Arithmetic	ADD	Numeric field or literal	Numeric field or literal
	DIV	Numeric field or literal	Numeric field or literal
	MULT	Numeric field or literal	Numeric field or literal
	MVR	Blank	Blank
	SUB	Numeric field or literal	Numeric field or literal
	Z-ADD	Blank	Numeric field or literal
	Z-SUB	Blank	Numeric field or literal
Array	LOKUP	Field or literal	Array name
	MOVEA	Blank	Field, literal, or array name
	SORTA	Blank	Array name
	XFOOT	Blank	Array name
Branching	GOTO	Blank	Label of a TAG operation or ENDSR operation
	EXSR	Blank	Subroutine name
	TAG	Label	Blank
Comparing and Testing	COMP	Field or literal	Field, literal, table of fields, or table of literals
	RANGE	Field or literal	Field or literal
Debugging	DEBUG	Field, literal, or blank	Blank
Indicator	SETOF	Blank	Blank
	SETON	Blank	Blank
Output/Input	GET	Blank	Master file name
	GETNH	Blank	Transaction file name or blank
	GETNR	Blank	Transaction file name or blank
	GETPH	Blank	Transaction file name or blank
	GETPR	Blank	Transaction file name or blank
	IMSG	Blank	MIC or message text
	MSG	Blank	MIC or message text
	PUT	Blank	Transaction file name or master file name
	PUTN	Blank	Master file name
	PUTS	Blank	Display format name
	TIME	Blank	Blank
Move	MOVE	Blank	Field, literal, or *BLANK
	MOVEL	Blank	Field or literal
Subroutine	BEGSR	Subroutine name	Blank
	ENDSR	Label or blank	Blank
	PRTY	Blank	0, 1, 2, or 3

Figure 17-1. Entries for Factor 1 and Factor 2

Operation (Columns 28 through 32)

Columns 28 through 32 specify the operation to be done using factor 1, factor 2, and/or the result field. The operation code must begin in column 28.

Figure 17-1 is a summary of operation codes and the entries for factor 1 and factor 2. Figure 17-2 is a summary of all the entries for operation codes. Chapter 7 contains examples of using operations; Chapter 18 contains descriptions of all the operation codes.

Table or Message Text (Columns 33 through 52)

For a COMP operation, you can code a table of values (field names or literals). Table elements must be separated with a semicolon. You can continue a table to columns 7 through 52 of the next line if you code a nonblank character in column 53.

For a MSG or IMSG operation, you can code message text enclosed in apostrophes. You can continue the message text to columns 7 through 52 of the next line if you code a nonblank character in column 53.

Literal (Columns 43 through 52)

A literal is the actual data used in an operation, rather than the field name representing that data. A literal can be alphameric or numeric.

The following rules apply to each **alphameric** literal:

- Any combination of characters and blanks can be used.
- The maximum length is 8 characters including blanks.
- It must be enclosed with apostrophes (').
- An apostrophe in the literal is represented by two apostrophes. For example, the literal O'CLOCK is coded as 'O''CLOCK'.
- It cannot be used for arithmetic operations.
- It must be left-adjusted.

The following are examples of alphameric literals:

'512% DT.'
'February'
'O''Clock'
'''82'

The following rules apply to each *numeric* literal:

- It consists of any combination of the digits 0 through 9. A decimal point, comma, or a sign (+ or -) can also be included, depending on the edit code specified in column 43 of the J-specification and column 25 of the D-specification.
- It must be left-adjusted.
- The sign (+ or -), if present, must be the leftmost character. WSU treats a numeric literal that is not signed as a positive number.

- The maximum total length is 10 digits including sign and decimal point.
- Embedded blanks are not allowed.
- It must *not* be enclosed in apostrophes (').
- It is used in the same way as a numeric field.

The following are examples of numeric literals:

12500
125.00
.001256789
-.01256789

For a RANGE operation, a numeric literal (up to 10 digits), an alphameric literal (up to 8 characters), or the name of a field upon which the operation is done can be coded in columns 43 through 52.

						Result Field				
Type of Operation	Operation Code (28-32)	Processing Function (7-8)	Conditioning Indicators (9-17)	Factor 1 (18-27)	Factor 2 (33-42)	Name (43-48)	Length (49-51)	Decimal Position (52)	Half-Adjust (53)	Resulting Indicators[1] (54-59)
Arithmetic	ADD	O	O	R	R	R	O	O	O	O
	DIV	O	O	R	R	R	O	O	O[2]	O
	MULT	O	O	R	R	R	O	O	O	O
	MVR	O	O	B	B	R	O	O	B	O
	SUB	O	O	R	R	R	O	O	O	O
	Z-ADD	O	O	B	R	R	O	O	O	O
	Z-SUB	O	O	B	R	R	O	O	O	O
Array	LOKUP	O	O	R	R	B	B	B	B	R
	MOVEA	O	O	B	R	R	O	O	B	B
	SORTA	O	O	B	R	B	B	B	B	B
	XFOOT	O	O	B	R	R	O	O	O	O
Branching	GOTO	O	O	B	R	B	B	B	B	B
	EXSR	O	O	B	R	B	B	B	B	B
	TAG	O	B	R	B	B	B	B	B	B
Comparing and Testing	COMP	O	O	R	R[3]	▨	▨	▨	▨	R
	RANGE	O	O	R	R	R	O	O	B	R
Debugging	DEBUG	O	O	O	B	O	B	B	B	B
Indicators	SETOF	O	O	B	B	B	B	B	B	R
	SETON	O	O	B	B	B	B	B	B	R
Output/Input	GET	O	O	B	R	B	B	B	B	O[4]
	GETNH	O	O	B	O[5]	B	B	B	B	O[4]
	GETNR	O	O	B	O[5]	B	B	B	B	O[4]
	GETPH	O	O	B	O[5]	B	B	B	B	O[4]
	GETPR	O	O	B	O[5]	B	B	B	B	O[4]
	IMSG	O	O	B	R[6]	▨	▨	▨	▨	O
	MSG	O	O	B	R[6]	▨	▨	▨	▨	O
	PUT	O	O	B	R	B	B	B	B	O[7]
	PUTN	O	O	B	R	B	B	B	B	R[8]
	PUTS	O	O	B	R	B	B	B	B	B
	TIME	O	O	B	B	R	O	O	B	B
Move	MOVE	O	O	B	R	R	O	O	B	B
	MOVEL	O	O	B	R	R	O	O	B	B
Subroutine	BEGSR	O	B	R	B	B	B	B	B	B
	ENDSR	O	B	O	B	B	B	B	B	B
	PRTY	O	B	B	R[9]	B	B	B	B	B

O = optional; R = required; B = blank

---

[1] Resulting indicators must not be coded on a line that does not have an operation code.

[2] MVR cannot follow a DIV operation that specified half-adjust.

[3] Factor 2 can be a field, literal, or table of values. Column 53 is used to indicate continuation of a table.

[4] You can code an indicator in columns 54 and 55 that turns on when WSU cannot find a record to read or cannot identify a record. Columns 56 through 59 must be blank.

[5] Factor 2 can be blank or the name of the transaction file. If factor 2 is blank, WSU assumes the name of the transaction file.

[6] Factor 2 can be as many as 64 characters of message text or a 4-digit MIC. Column 53 is used to indicate continuation of message text.

[7] You can code the type of record to be written to the transaction file in columns 54 and 55; columns 56 through 59 must be blank.

[8] You must code the type of record to be added to the master file in columns 54 and 55; columns 56 through 59 must be blank.

[9] Factor 2 must be 0, 1, 2, or 3, and must be left-adjusted.

Figure 17-2. Operation Codes

Result Field (Columns 43 through 53)

The *Result Field* area on the C-specification contains information about the results of the operation that was named in columns 28 through 32.

The result field consists of four entries:

- Name
- Length
- Decimal positions
- Half-adjust.

Result Field Name (Columns 43 through 48)

This entry names either the field (or array) that will contain the results of the operation coded in columns 28 through 32, or (as for the RANGE operation) the field or literal upon which the operation is done. You can define a result field by coding a new field name, or you can use the name of a field that you have defined elsewhere in the program. During program running, WSU reserves storage in a field area for copies of any field that you define. For a field defined elsewhere in a program, columns 49 through 52 can be blank; however, any entries in columns 49 through 52 must agree with all other definitions of the field in the program.

A job field can be defined as the result field name. Only one copy of a job field is generated per program, rather than one copy per each display station that uses the program. Job field names must begin with & and must be left-adjusted. The second character must be alphabetic; the remaining characters (up to 4) must be alphameric. Job field names cannot contain blanks.

Result Field Length (Columns 49 through 51)

To specify the length of the result field, you can enter a number from **1 through n**. The maximum length, *n*, is 15 for a numeric field and 256 for an alphameric field. The length must be right-adjusted.

If the field is defined elsewhere in this program (on the I-specifications in this program's file definition or on another C-specification in the program), leave this entry **blank**.

If any **other entry** is made, WSU assumes a blank.

Those result fields defined as packed fields on I-specifications can have their lengths redefined by this entry. The result field length must allow for the unpacked length of a field. See Chapter 7, *Coding Processing*, for examples of using result fields and result field lengths.

Result Field Decimal Positions (Column 52)

To show the number of decimal positions wanted to the right of the decimal in a numeric result field, you can enter a number from **0 through 9**. If the numeric result field contains no decimal positions, enter a 0 (zero).

The number of decimal positions must never be greater than the length of the field. The number can, however, be larger or smaller than the number of decimal positions that actually result from an operation. If the number of decimal positions specified is greater than the number of decimal places that actually result from an operation, zeros fill in to the right. If the number specified is smaller than the number that results from the operation, the rightmost digits drop.

Figure 17-3 shows how the contents of a numeric result field after a multiplication operation can change according to the decimal positions (column 52) and field length (columns 49 through 51) specifications.

If this is an alphameric result field or a numeric result field defined by an I-specification or C-specification elsewhere in the program, leave this column **blank**. The decimal-positions entry must be blank if the field-length entry (columns 49 through 51) is blank.

If any **other entry** is made, WSU assumes an entry of 0 if the field length (columns 49 through 51) is not blank and assumes blank if the field length is blank.

Multiplication: 98.76 x 1.234 = 121.86984

Decimal Positions for Result Field (column 52)	**Result Field Length (columns 49 through 51)**									
	10	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
9	1.869840000	.869840000	▨	▨	▨	▨	▨	▨	▨	▨
8	21.86984000	1.86984000	.86984000	▨	▨	▨	▨	▨	▨	▨
7	121.8698400	21.8698400	1.8698400	.8698400	▨	▨	▨	▨	▨	▨
6	0121.869840	121.869840	21.869840	1.869840	.869840	▨	▨	▨	▨	▨
5	00121.86984	0121.86984	121.86984	21.86984	1.86984	.86984	▨	▨	▨	▨
4	000121.8698	00121.8698	0121.8698	121.8698	21.8698	1.8698	.8698	▨	▨	▨
3	0000121.869	000121.869	00121.869	0121.869	121.869	21.869	1.869	.869	▨	▨
2	00000121.86	0000121.86	000121.86	00121.86	0121.86	121.86	21.86	1.86	.86	▨
1	000000121.8	00000121.8	0000121.8	000121.8	00121.8	0121.8	121.8	21.8	1.8	.8
0	0000000121	000000121	00000121	0000121	000121	00121	0121	121	21	1

▨ Not permitted ░ Permitted but inaccurate ☐ Recommended

Figure 17-3. Result Field Contents Based on Various Field Lengths and Decimal Positions

Result Field Half-Adjust (Column 53)

This entry indicates whether or not the contents of the result field are half-adjusted (rounded).

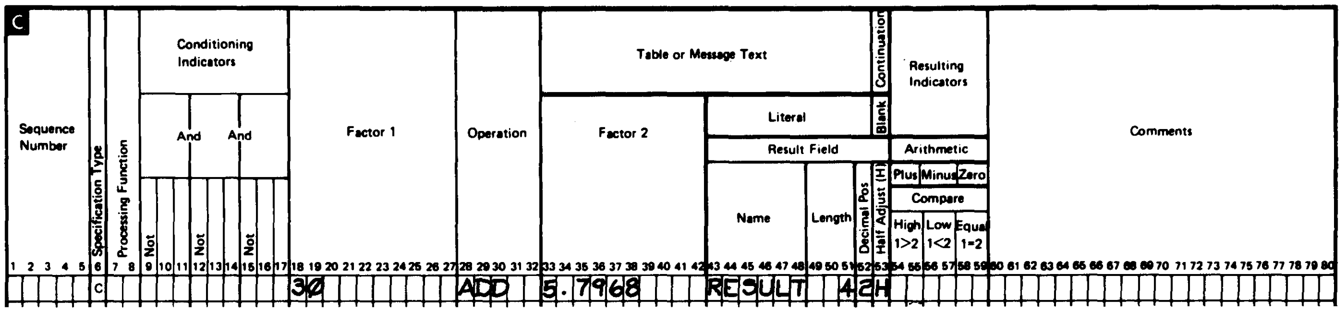
If you do not want the field half-adjusted, leave the column **blank**.

If you want the field half-adjusted, enter an **H**.

If any **other entry** is made, WSU assumes an entry of H.

Half-adjusting adds 5 (-5 if the result is negative) to the number at the right of the last decimal position specified for the field. All decimal positions to the right of the position specified for that field drop (Figure 17-4).

The half-adjust entry is allowed with all arithmetic operations except MVR. Also, if half-adjust is specified for a DIV operation, an MVR operation should not follow it.



Specification Type	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Pos	Half Adjust (H)
C	30	ADD	5.7968	RESULT	4	2	H

Figure 17-4. Half-Adjust Operation

Use of Column 53 to Continue Message Text

Column 53 can also be used to continue message text for an IMSG or MSG operation or to continue a table for a COMP operation. To allow continuation, you must code any character except a blank in this column.

Resulting Indicators (Columns 54 through 59)

Use columns 54 and 55, 56 and 57, and 58 and 59 to indicate the results of operations. You can code these columns only on C-specification lines that specify operation codes.

Before an operation is done, all its resulting indicators should be set off. After the operation is done, WSU checks the result and sets on the appropriate indicators.

Figure 17-5 shows resulting indicators that can be used with each operation. Refer to Chapter 3 for descriptions of the indicators.

Plus or High 1 > 2 (Columns 54 and 55)

An indicator in these columns turns on if:

- The result of an arithmetic operation is positive.
- Factor 1 is greater than factor 2 in a COMP operation.
- One of the get operations was not successful.
- Factor 1 is greater than the higher field in a RANGE operation.

Minus or Low 1 < 2 (Columns 56 and 57)

An indicator in these columns turns on if:

- The result of an arithmetic operation is negative.
- Factor 1 is less than factor 2 in a COMP operation.
- Factor 1 is not equal to any entry in a table in a COMP operation.
- Factor 1 is less than the lowest field in a RANGE operation.

Zero or Equal 1=2 (Columns 58 and 59)

An indicator in these columns turns on if:

- The result of an arithmetic operation is zero.
- Factor 1 equals at least one entry in a table in a COMP operation.
- Factor 1 equals factor 2 in a COMP operation.
- Factor 1 is equal to one value or between the two values in a RANGE operation.

	Resulting Indicators															
Operation	**AC**	**AE**	**EJ**	**ES**	**EW**	**IP**	**IS**	**JA-JN JP-JY**	**KG-KL KQ-KY**	**PG**	**RC**	**RP**	**RS**	**SA-SN SP-SY**	**U1-U8**	**01-89**
ADD	x	x	x	x	x			x	x					x	x	x
COMP	x	x	x	x	x			x	x					x	x	x
DIV	x	x	x	x	x			x	x					x	x	x
GET																x
GETNH																x
GETNR																x
GETPH																x
GETPR																x
IMSG	x	x	x	x	x			x	x					x	x	x
MSG	x	x	x	x	x			x	x					x	x	x
MULT	x	x	x	x	x			x	x					x	x	x
MVR	x	x	x	x	x			x	x					x	x	x
PUT																x
PUTN																x
RANGE	x	x	x	x	x			x	x					x	x	x
SETOF	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SETON	x	x	x	x	x			x	x	x	x		x	x	x	x
SUB	x	x	x	x	x			x	x					x	x	x
Z-ADD	x	x	x	x	x			x	x					x	x	x
Z-SUB	x	x	x	x	x			x	x					x	x	x

Notes:

1. An x in the above figure specifies a resulting indicator that can be used by the operation.
2. BEGSR, DEBUG, ENDSR, EXSR, GOTO, LOKUP, MOVEA, MOVE, MOVEL, PRTY, PUTS, SORTA, TAG, and TIME do not allow resulting indicators.
3. Indicators CG, DL, IJ, IN, IW, RR, RU, and RV are turned on and off only by WSU.

Figure 17-5. Using Resulting Indicators

Additional Uses of Resulting Indicators

You can specify indicators in columns 54 through 59 for the MSG, IMSG, SETON, and SETOF operations. For MSG, IMSG, and SETON, the specified indicators turn on when the operation runs. For SETOF, the specified indicators turn off when the operation runs.

You can specify one indicator in columns 54 and 55 for a PUT operation. The indicator specifies (in enter mode) the type of record to write to the transaction file. You can specify one indicator in columns 54 and 55 for a PUTN operation. The indicator specifies the type of record to write to a master file.

Refer to Figure 17-5 for a chart showing the operations and the resulting indicators that can be used.

Comments (Columns 60 through 80)

Comments that you code do not affect the WSU program, but are used to make your program easier to understand.

Chapter 18. Operation Codes

Columns 28 through 32 on the *WSU Processing Specifications* sheet specify the operations to be done using factor 1, factor 2, and a result field. The operation code must begin in column 28. Operations are also discussed in Chapter 7, *Coding Processing,* and Chapter 17 *C-(Processing) Specifications.*

ADD

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Required*	ADD	*Required*	*Required*	*Optional*

Factor 1: Numeric field or literal
Factor 2: Numeric field or literal

Adds factor 1 to factor 2 and places the sum in the result field.

Factor 1 and factor 2 do not change, unless either is also the result field.

BEGSR (Begin Subroutine)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Blank	*Required*	BEGSR	*Blank*	*Blank*	*Blank*

Factor 1: Subroutine name
Factor 2: Blank

Labels the beginning of a subroutine.

Factor 1 can be 1 to 6 characters long. The first character must be alphabetic; the remaining characters must be alphameric. No two subroutines can have the same name.

Refer to Chapter 7, *Coding Processing,* for an example of coding the BEGSR operation.

Notes:

1. You must code an ENDSR operation after a BEGSR operation before you can code another BEGSR operation.
2. The subroutine name in factor 1 must not be the same as the name in factor 1 of a TAG or ENDSR operation or as any field name defined in the program.

COMP (Compare)

Conditioning Indicators	Factor 1	Operation	Table or Message Text	Resulting Indicators
Optional	*Required*	COMP	*Required*	*Required*

Factor 1: Field or literal
Factor 2: Field, literal, table of fields, or table of literals

Compares factor 1 to factor 2 and sets a resulting indicator to show if the comparison was high, low, or equal.

You can code conditioning indicators in columns 9 through 17.

For factor 2, you can code a table of values in columns 33 through 52. Table entries must be either field names or literals but cannot be a combination of the two. Table entries must be either all alphameric or all numeric. You must separate table elements with a semicolon.

You can continue a table to columns 7 through 52 of a second specification line by coding any character except a blank in column 53 of the first specification line. The first line of the table must end with a semicolon so that the second line of the table begins with a left-adjusted table entry. Do not begin the second line of the table with an asterisk (*) in column 7 because WSU treats such lines as comments.

A table of values cannot be continued to a third line.

Refer to Chapter 7, *Coding Processing,* for examples of the COMP operation.

DEBUG

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Optional*	DEBUG	*Blank*	*Optional*	*Blank*

Factor 1: Field, literal, or blank
Factor 2: Blank

Provides a means of detecting errors while the program is running.

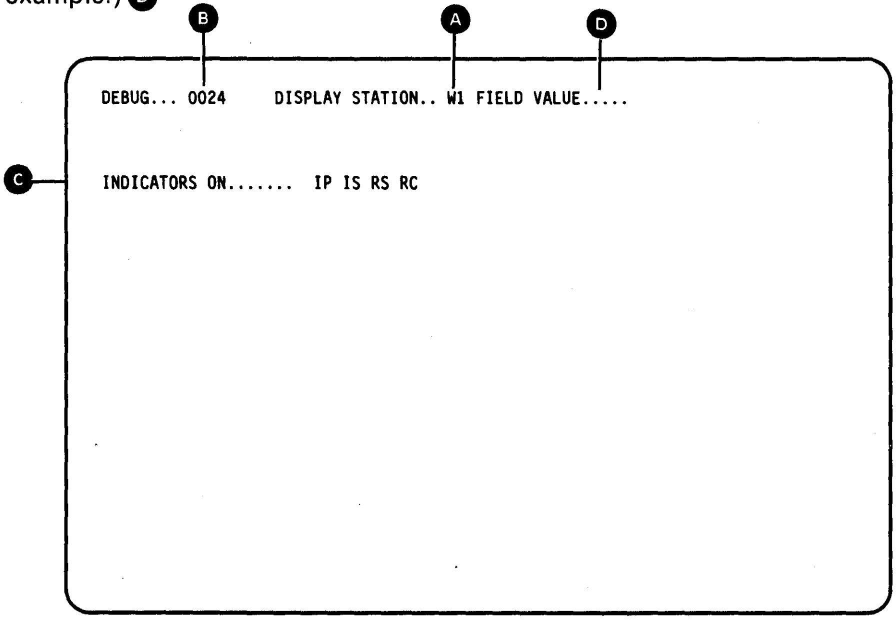
The length of the factor 1 value, if used, must not exceed 8 characters. You can code indicators in columns 9 through 17 and a result field in columns 43 through 48. All other entries must be blank.

During program generation, WSU adds a DEBUG-YES parameter to the WSX statement in the WSU-generated procedure if one or more DEBUG operations occur in the program. You can modify the WSU-generated procedure to set the debugging function on or off before the WSU program runs by specifying YES (on) or NO (off) for the DEBUG parameter. Refer to *Modifying the WSU-Generated*

Procedure in Chapter 8 for an explanation on how to change the DEBUG parameter.

If DEBUG-YES is specified in the procedure that calls the program, WSU saves the current display and shows a debugging display each time a DEBUG operation is encountered during program running. The debugging display contains:

- The display station's ID.Ⓐ
- The contents of the field or literal in factor 1, or the statement number of the DEBUG operation if factor 1 is blank.Ⓑ
- A list of indicators that are on, or the word NONE if no indicators are on. Ⓒ
- The contents of the result field, if a result is specified. (Blank in example.)Ⓓ



```
DEBUG... 0024    DISPLAY STATION.. W1 FIELD VALUE.....


INDICATORS ON.......  IP IS RS RC
```

You can press the Print key to print the display. When you press the Enter key, the saved display (display before the DEBUG display appeared) is shown again, and processing resumes at the operation following the DEBUG operation.

Note: DEBUG operations occupy main storage. Therefore, after the program is debugged, removing the DEBUG operations and generating the program again might improve performance.

DIV (Divide)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Required*	DIV	*Required*	*Required*	*Required*

Factor 1: Numeric field or literal
Factor 2: Numeric field or literal

Divides factor 1 by factor 2 and places the quotient in the result field.

Factor 1 and factor 2 do not change unless either is also the result field. If factor 1 is zero, the result of the divide operation is zero. If factor 2 is zero, WSU issues a message and abnormally ends the work session. Any remainder resulting from a divide operation is lost unless you code the move remainder operation (MVR) as the next operation.

ENDSR (End Subroutine)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Blank	*Optional*	ENDSR	*Blank*	*Blank*	*Blank*

Factor 1: Label
Factor 2: Blank

Defines the end of a subroutine and returns control to the instruction after the EXSR operation that called the subroutine.

Factor 1 can be 1 to 6 characters long. The first character must be alphabetic; the remaining characters must be alphameric. This label serves as a point which you can branch to by a GOTO statement within the subroutine and, therefore, cannot be the same as any of the following:

- Factor 1 of a TAG operation
- Factor 1 of another ENDSR operation
- Any other field name defined in the program.

Conditioning indicators (columns 9 through 17) and factor 2 cannot be coded for an ENDSR operation.

Refer to Chapter 7, *Coding Processing,* for an example of coding the ENDSR operation.

EXSR (Execute Subroutine)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	EXSR	*Required*	*Blank*	*Blank*

Factor 1: Blank
Factor 2: Subroutine name

Branches to and runs the subroutine named in factor 2.

Indicators in columns 9 through 17 can condition an EXSR operation. Factor 2 must be the same name as factor 1 of the corresponding BEGSR instruction.

An EXSR operation can appear anywhere in the program except within a subroutine. This means that a subroutine cannot call itself or another subroutine. Whenever an EXSR operation appears, the subroutine runs. After operations in the subroutine complete, control returns to the operation following the EXSR operation.

Refer to Chapter 7, *Coding Processing,* for an example of coding the EXSR operation.

GET (Get a Master File Record)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	GET	*Required*	*Blank*	*Optional*

Factor 1: Blank
Factor 2: Master file name

Reads a master file record so that it can be used in the WSU program.

You can code conditioning indicators in columns 9 through 17. Also, you can code a record-not-found indicator in columns 54 and 55 that turns on when WSU cannot recognize the type of record read. Columns 43 through 53 and 56 through 59 must be blank.

WSU forms the key to read a master file record from the GET fields coded in columns 31 through 48 of the M-specification. WSU turns on the indicator for the type of record read and turns off all other record-identifying indicators for the master file. If WSU can recognize the record type, it places the fields for that record type into the field area.

WSU provides a way to detect output/input errors that occur during a GET operation by allowing you to code an error indicator in columns 61 and 62 of the M-specification for the master file. Each time WSU starts a GET operation, that error indicator is set off, and the *ERROR reserved field is set to zero. If an error occurs during the GET operation, the error indicator is set on, and an error code is placed in the *ERROR field. Refer to Chapter 3 for information about *ERROR and the error codes that this field can contain. Refer to Chapter 15 for a description of the Error Indicator entry on the M-specification.

If you code different not-found indicators on the M-specification and the C-specification, WSU uses the indicator on the C-specification for the GET operation. The not-found indicator on the M-specification does not change and might reflect the result of a previous GET operation. Refer to Chapter 15 for a description of the Not-Found Indicator entry on the M-specification.

A record-not-found condition is treated as an error only if not-found indicators are omitted both from the GET operation and from the M-specification. Therefore, any error indicator in columns 61 and 62 of the M-specification is used when a not-found condition occurs and columns 54 and 55 of the GET operation and columns 49 and 50 of the M-specification for the master file are blank.

If a not-found condition or an output/input error occurs and indicators have not been coded in the program to reflect the occurrence, the system operator receives WSU message 0803 and an error code. The operator should refer to the *Utilities Messages* manual for a description of the error.

GETNH (Get the Next Header Record in the Transaction File)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	GETNH	*Optional*	*Blank*	*Optional*

Factor 1: Blank
Factor 2: Transaction file name or blank

Reads the next header record from the transaction file.

If factor 2 is blank, WSU assumes the transaction file name. You can code conditioning indicators in columns 9 through 17. Also, you can code a record-not-found indicator in columns 54 and 55. This indicator turns on when WSU tries to read beyond the last record in the chain or when WSU cannot recognize the type of record read.

A record-not-found condition occurs when WSU, trying to read the next header record in the chain, reads beyond the last record in the chain. WSU turns on the indicator for the type of record read and turns off all other record-identifying indicators for the transaction file as well as the RU indicator. A record-not-found condition also occurs when WSU cannot recognize the record type.

If WSU can recognize the record type, it places the fields for that record type into the field area and places the relative record number of the record in the *RLRU reserved field.

WSU provides a way to detect output/input errors that might occur during a GETNH operation by allowing you to code an error indicator in columns 61 and 62 of the T-specification. Each time WSU starts the GETNH operation, that error indicator is set off, and the *ERROR reserved field is set to zero. If an error occurs during the GETNH operation, the error indicator is set on, and an error code is placed in *ERROR. Refer to Chapter 3 for information about *ERROR and the error codes that this field can contain. Refer to Chapter 15 for a description of the Error Indicator entry on the T-specification.

WSU also provides ways to detect record-not-found conditions that might occur during a GETNH operation by allowing you to code a not-found indicator in columns 54 and 55 of the C-specification or in columns 49 and 50 of the T-specification.

If you code different not-found indicators on the T-specification and the C-specification, the indicator on the C-specification is used for the GETNH operation. The not-found indicator on the T-specification does not change and might reflect the result of a previous GETNH operation. Refer to Chapter 15 for a description of the Not-Found Indicator entry on the T-specification.

A record-not-found condition is treated as an error only if not-found indicators are omitted from the GETNH operation and from the T-specification. Therefore, any error indicator in columns 61 and 62 of the T-specification is used when a not-found condition occurs and columns 54 and 55 of the GETNH operation and columns 49 and 50 of the T-specification are blank.

If a not-found condition or an output/input error occurs and indicators have not been coded in the program to reflect the occurrence, the system operator receives WSU message 0803 and an error code. The operator should refer to the *Utilities Messages* manual for a description of the error.

Notes:

1. The GETNH operation causes a terminal error during program generation if the program does not specify a transaction file.

2. If you code a GETNH operation in your program and there are no header records in the file (columns 55 and 56 of the T-specification are blank), WSU issues a terminal message during program generation.

GETNR (Get the Next Record in the Transaction File)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	GETNR	*Optional*	*Blank*	*Optional*

Factor 1: Blank
Factor 2: Transaction file name or blank

Reads the next logical record from the transaction file.

If factor 2 is blank, WSU assumes the transaction file name. You can code conditioning indicators in columns 9 through 17. Also, you can code a record-not-found indicator in columns 54 and 55. This indicator turns on when WSU tries to read beyond the last record in a chain or when WSU cannot recognize the type of record read.

A record-not-found condition occurs when WSU, trying to read the next record in the chain, reads beyond the last record in the chain. WSU turns on the indicator for the type of record read and turns off all other record identifying indicators for the transaction file. A record-not-found condition also occurs when WSU cannot recognize the record type.

If WSU can recognize the record type, it places the fields for that record type into the field area and places the relative record number of the record in the *RLRU reserved field.

WSU provides a way to detect output/input errors that might occur during a GETNR operation by allowing you to code an error indicator in columns 61 and 62 of the T-specification. Each time WSU starts the GETNR operation, that error indicator is set off, and the *ERROR reserved field is set to zero. If an error occurs during the GETNR operation, the error indicator is set on, and an error code is placed in *ERROR. Refer to Chapter 3 for information about *ERROR and the error codes that this field can contain. Refer to Chapter 15 for a description of the Error Indicator entry on the T-specification.

WSU also provides ways to detect record-not-found conditions that might occur during a GETNR operation by allowing you to code a not-found indicator in columns 54 and 55 of the C-specification or in columns 49 and 50 of the T-specification.

If you code different not-found indicators on the T-specification and the C-specification, the indicator on the C-specification is used for the GETNR operation. The not-found indicator on the T-specification does not change and might reflect the result of a previous GETNR operation. Refer to Chapter 15 for a description of the Not-Found Indicator entry on the T-specification.

A record-not-found condition is treated as an error only if not-found indicators are omitted from the GETNR operation and from the T-specification. Therefore, any error indicator in columns 61 and 62 of the T-specification is used when a not-found condition occurs and columns 54 and 55 of the GETNR operation and columns 49 and 50 of the T-specification are blank.

If a not-found condition or an output/input error occurs and indicators have not been coded in the program to reflect the occurrence, the system operator receives WSU message 0803 and an error code. The operator should refer to the *Utilities Messages* manual for a description of the error.

Notes:

1. The GETNR operation causes a terminal error during program generation if the program does not specify a transaction file.
2. The GETNR operation cannot read records that are being inserted in the transaction file until the processing of the insert display ends.

GETPH (Get the Previous Header Record in the Transaction File)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	GETPH	*Optional*	*Blank*	*Optional*

Factor 1: Blank
Factor 2: Transaction file name or blank

Reads the previous logical header record from the transaction file.

If factor 2 is blank, WSU assumes the transaction file name. You can code conditioning indicators in columns 9 through 17. Also, you can code a record-not-found indicator in columns 54 and 55. This indicator turns on when WSU tries to read ahead of the first record in a chain or when WSU cannot recognize the type of record read.

A record-not-found condition occurs when WSU, trying to read the previous header record in the chain, reads ahead of the first record in the chain. WSU turns on the indicator for the type of record read and turns off all other record-identifying indicators for the transaction file. A record-not-found condition also occurs when WSU cannot recognize the record type.

If WSU can recognize the record type, it places the fields for that record type into the field area and places the relative record number of the record in the *RLRU reserved field.

WSU provides a way to detect output/input errors that might occur during a GETPH operation by allowing you to code an error indicator in columns 61 and 62 of the T-specification. Each time WSU starts the GETPH operation, that error indicator is set off, and the *ERROR reserved field is set to zero. If an error occurs during the GETPH operation, the error indicator is set on, and an error code is placed in *ERROR. Refer to Chapter 3 for information about *ERROR and the error codes that this field can contain. Refer to Chapter 4 for a description of the Error Indicator entry on the T-specification.

WSU also provides ways to detect record-not-found conditions that might occur during a GETPH operation by allowing you to code a not-found indicator in columns 54 and 55 of the C-specification or in columns 49 and 50 of the T-specification.

If you code different not-found indicators on the T-specification and the C-specification, the indicator on the C-specification is used for the GETPH operation. The not-found indicator on the T-specification does not change and might reflect the result of a previous GETPH operation. Refer to Chapter 15 for a description of the Not-Found Indicator entry on the T-specification.

A record-not-found condition is treated as an error only if not-found indicators are omitted from the GETPH operation and from the T-specification. Therefore, any error indicator in columns 61 and 62 of the T-specification is used when a not-found condition occurs and columns 54 and 55 of the GETPH operation and columns 49 and 50 of the T-specification are blank.

If a not-found condition or an output/input error occurs and indicators have not been coded in the program to reflect the occurrence, the system operator receives WSU message 0803 and an error code. The operator should refer to the *Utilities Messages* manual for a description of the error.

Notes:

1. The GETPH operation causes a terminal error during program generation if the program does not specify a transaction file.

2. If you code a GETPH operation in your program and there are no header records in the file (columns 55 and 56 of the T-specification are blank), WSU issues a terminal message during program generation.

GETPR (Get the Previous Record in the Transaction File)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	GETPR	*Optional*	*Blank*	*Optional*

Factor 1: Blank
Factor 2: Transaction file name or blank

Reads the previous logical record from the transaction file.

If factor 2 is blank, WSU assumes the transaction file name. You can code conditioning indicators in columns 9 through 17. Also, you can code a record-not-found indicator in columns 54 and 55. This indicator turns on when WSU tries to read ahead of the first record in the chain or when WSU cannot recognize the type of record read.

A record-not-found condition occurs when WSU, trying to read the previous record in the chain, reads ahead of the first record in the chain. WSU turns on the indicator for the type of record read and turns off all other record-identifying indicators for the transaction file. A record-not-found condition also occurs when WSU cannot recognize the record type.

If WSU can recognize the record type, it places the fields for that record type into the field area and places the relative record number of the record in the *RLRU reserved field.

WSU provides a way to detect output/input errors that occur during a GETPR operation by allowing you to code an error indicator in columns 61 and 62 of the T-specification. Each time WSU starts the GETPR operation, that error indicator is set off, and the *ERROR reserved field is set to zero. If an error occurs during the GETPR operation, the error indicator is set on, and an error code is placed in *ERROR. Refer to Chapter 3 for information about *ERROR and the error codes that this field can contain. Refer to Chapter 15 for a description of the Error Indicator entry on the T-specification.

WSU also provides ways to detect record-not-found conditions that might occur during a GETPR operation by allowing you to code a not-found indicator in columns 54 and 55 of the C-specification or in columns 49 and 50 of the T-specification.

If you code different not-found indicators on the T-specification and the C-specification, the indicator on the C-specification is used for the GETPR operation. The not-found indicator on the T-specification does not change and might reflect the result of a previous GETPR operation. Refer to Chapter 15 for a description of the Not-Found Indicator entry on the T-specification.

A record-not-found condition is treated as an error only if not-found indicators are both omitted from the GETPR operation and from the T-specification. Therefore, any error indicator in columns 61 and 62 of the T-specification is used when a not-found condition occurs and columns 54 and 55 of the GETPR operation and columns 49 and 50 of the T-specification are blank.

If a not-found condition or an output/input error occurs and indicators have not been coded in the program to reflect the occurrence, the system operator receives WSU message 0803 and an error code. The operator should refer to the *Utilities Messages* manual for a description of the error.

Notes:

1. The GETPR operation causes a terminal error during program generation if the program does not specify a transaction file.
2. The GETPR operation cannot read records that are being inserted in the transaction file until the processing of the insert display ends.

GOTO

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	GOTO	*Required*	*Blank*	*Blank*

Factor 1: Blank
Factor 2: Label of a TAG operation or an ENDSR operation

Skips instructions and goes to a TAG or ENDSR statement in the program.

Factor 2 must contain the name of the statement branched to (identified by the label on a TAG or ENDSR line). The label in factor 2 must be left-adjusted.

You can condition the GOTO operation with indicators in columns 9 through 17.

IMSG (Display an Information Message)

Conditioning Indicators	Factor 1	Operation	Table or Message Text	Resulting Indicators
Optional	*Blank*	IMSG	*Required*	*Optional*

Factor 1: Blank
Factor 2: MIC or message text

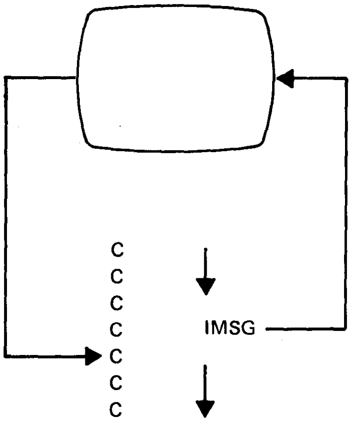
Displays a user-defined message on the bottom line of the current display.

If resulting indicators are specified by the operation, those indicators turn on. The current display is not reissued; the only change to it is the message. The processing of the C-specifications stops, and the operator cannot type data into any field on the display.

To clear the message, the operator must press the Error Reset key. To resume processing at the C-specification that follows the IMSG operation, the operator must press the Enter key, an enabled function key, or an enabled user command key.

If an IMSG operation is performed in processing that has no associated display (ES processing, for example), the message is displayed on the bottom line of the current display. The operator cannot type data into any field on the display. If an IMSG operation is performed while the WSU display is shown, the screen is cleared before the message is displayed.

If an IMSG operation is performed in preprocessing before a PUTS for the display occurs, the entire current display is shown and the message is displayed on its bottom line.



You can code conditioning indicators in columns 9 through 17 and resulting indicators in columns 54 through 59. The indicators in columns 54 through 59 turn on when an IMSG operation runs, and you can use these indicators to condition later operations, but not to condition display attributes or field attributes.

Columns 33 through 52 can contain message text or a MIC. Message text can be alphameric, Katakana, or ideographic data. Chapter 20, *Ideographic Considerations* has more information on Katakana and ideographic data. Message text must be enclosed in apostrophes. Code two consecutive apostrophes to represent one apostrophe in text.

You can continue text to columns 7 through 52 of the following C-specification line by coding any character except a blank in column 53 of the line you want to continue and coding the remainder of the text (left-adjusted) in columns 7 through 52 of the next line. Column 7 of the continuation line must not be an asterisk (*) because WSU treats such lines as comments. WSU allows 64 positions of message text. See Chapter 20, *Ideographic Considerations* for ideographic character considerations.

An MIC is a 4-digit number that references the appropriate message in your message member. Code the MIC in columns 33 through 36. Columns 23 through 30 of the J-specification must contain the name of your message member.

LOKUP (Look for Next Element of Array)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Required*	LOKUP	*Required*	*Blank*	*Required*

Factor 1: Field or literal
Factor 2: Array name

Searches factor 2 for an array element and turns on a resulting indicator if the search is successful.

Factor 1 is the search word (data for which you want to find a match in the array named). Factor 1, the search word, can be:

- An alphameric or numeric constant
- A field name
- An array element.

Resulting indicators are always used with a LOKUP operation. The indicators first specify the type of search to be made and then reflect the result of the search. The specified indicator turns on only if the search is successful.

Resulting indicators specify the type of search and reflect the result of the search in the following manner:

- A resulting indicator assigned to equal (columns 58 and 59) instructs the program to search for an entry in the array equal to the search word. The first equal entry found turns on the indicator assigned to the equal.
- An indicator assigned to low (columns 56 and 57) instructs the program to locate an entry in the array that is nearest to, yet lower in sequence, than the search word. The first such entry found turns on the indicator assigned to low.

- The indicator assigned to high (columns 54 and 55) instructs the program to find the entry that is nearest to, yet higher in sequence than, the search word. The first higher entry found turns on the indicator assigned to high.

At least one resulting indicator must be assigned, but no more than two can be used. Resulting indicators can be assigned to equal and high or to equal and low. The program searches for an entry that satisfies either condition; however, if no equal entry is found, the nearest lower or nearest higher entry is selected. If resulting indicators are assigned both to high and low, the indicator assigned to low is ignored.

When you use the LOKUP operation, remember:

- The search word and each array element must have the same length and the same format (alphameric or numeric).
- A search can be made for high, low, high and equal, or low and equal only if the array is in sequence.
- No resulting indicator turns on if the search is not successful.

To save processing time, you can start the LOKUP search at a particular element in the array. This type of search is indicated by additional entries in columns 33 through 42. Enter the name of the array to be searched in these columns followed by a comma and a numeric literal or by the name of a numeric field (with 0 decimal positions). The numeric literal or numeric field provides the number of the element at which the search is to start. This numeric literal or field is called the *index* because it points to a certain element in the array. All other columns are used as previously described for the normal LOKUP operation.

You can ensure that the elements of the array are in the proper sequence for a LOKUP operation by performing a SORTA operation.

Refer to Chapter 7, *Coding Processing*, for examples of coding the LOKUP operation.

MOVE (Move)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	MOVE	*Required*	*Required*	*Blank*

Factor 1: Blank
Factor 2: Field, literal, or *BLANK

Moves all or part of factor 2 into the result field.

Moving begins with the rightmost character of factor 2, which moves into the rightmost character of the result field. Factor 2 does not change. Resulting indicators cannot be used.

If factor 2 is longer than the result field, the excess leftmost characters of factor 2 do not move. If factor 2 is shorter than the result field, the extra leftmost characters in the result field do not change.

The MOVE operation can be used to change numeric data into alphameric data and alphameric data into numeric data. To change numeric data into alphameric data, code the name of the numeric field in factor 2 and the name of an alphameric field in the result field. To change alphameric data into numeric data, code the name of the alphameric field in factor 2 and the name of a numeric field in the result field.

When an alphameric field is moved into a numeric result field, the digit portion of each character is converted to its corresponding numeric character and then moved to the result field. Blanks are transferred as zeros. The zone portion of the rightmost alphameric character is converted to its corresponding sign and is moved to the rightmost position of the numeric field and becomes the sign of the field.

Refer to Chapter 7, *Coding Processing*, for examples of moving fields with the MOVE operation.

When the MOVE operation is used to move data into a numeric field, the decimal positions specified for the field in factor 2 are ignored. For example, if 1.00 is moved into a numeric field that has one decimal position, the result is 10.0.

You can blank an alphameric field by coding the MOVE operation with the *BLANK reserved field as factor 2 and the name of the field you want to blank as the result field.

MOVEA (Move Array)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	MOVEA	*Required*	*Required*	*Blank*

Factor 1: Blank
Factor 2: Field, literal, or array name

Transfers characters from factor 2 to the result field.

Factor 2 and the result field cannot reference the same array even if the array is indexed. The array or field referenced by a MOVEA operation must be alphameric.

The length of the move is determined by the shorter of the lengths of factor 2 and the result field. If factor 2 is longer than the result field, the excess rightmost characters of factor 2 are not moved; if the result field is longer than factor 2, the rightmost characters in the result field are unchanged.

The MOVEA operation makes it possible to:

- Move several consecutive array elements to a single field.
- Move a single field to several consecutive array elements.
- Move consecutive elements of one array to consecutive elements of another array.

Movement of data starts with the first element of an array if the array is not indexed or with the specified element of an array if the array is indexed. The movement of the data ends when the last array element is moved or filled or when the number of characters moved equals the length of the shorter field specified by factor 2 and the result field; therefore, the move could end in the middle of an array element.

If you use the MOVEA operation with *BLANK in factor 2, and an array in the result field, *BLANK fills the entire array.

If you use the MOVEA operation with *BLANK in factor 2, and an array element in the result field *BLANK fills the array from the referenced element to the end of the array.

OVEL (Move Left)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	MOVEL	*Required*	*Required*	*Blank*

Factor 1: Blank
Factor 2: Field or literal

Moves all or part of factor 2 into the result field.

Moving begins with the leftmost character of factor 2, which moves into the leftmost character of the result field. Factor 2 does not change. Factor 1 and resulting indicators must be blank.

If factor 2 is longer than the result field, the excess rightmost characters of factor 2 do not move. If factor 2 is shorter than the result field, the excess rightmost characters of the result field do not change.

The MOVEL operation can be used to change numeric data into alphameric data and alphameric data into numeric data. To change numeric data into alphameric data, code the name of the numeric field in factor 2 and name of an alphameric field in the result field. To change alphameric data into numeric data, code the name of the alphameric field in factor 2 and the name of a numeric field in the result field.

When an alphameric field is moved into a numeric result field, the digit portion of each character is converted to its corresponding numeric character and then moved to the result field. Blanks are transferred as zeros. When factor 2 is longer than the result field or the same length as the result field, the zone portion of the rightmost character of factor 2 is converted and used as the sign of the result field, whether or not the rightmost character is included in the move. Chapter 7, *Coding Processing,* contains examples of moving fields with the MOVEL operation.

When the MOVEL operation is used to move data into a numeric field, the decimal positions specified for the field in factor 2 are ignored. For example, if 1.00 is moved into a numeric field that has one decimal position, the result is 10.0.

A summary of the rules for MOVEL operations for three conditions based on the relationships of field lengths is as follows:

If factor 2 is the same length as the result field and:

1. If factor 2 is numeric and the result field is either numeric or alphameric, the sign is moved with the rightmost position.
2. If factor 2 is alphameric and the result field is numeric, a minus zone is moved into the rightmost position of the result field if the zone from the rightmost position of factor 2 is a D (minus zone). However, if the zone from the rightmost position of factor 2 is not a D, a positive zone is moved into the rightmost position of the result field. Digit portions are converted to their corresponding numeric characters.
3. If factor 2 and the result field are alphameric, all characters are moved.

If factor 2 is longer than the result field and:

1. If factor 2 and the result field are numeric, the sign from the rightmost position of factor 2 is moved into the rightmost position of the result field.
2. If factor 2 is numeric and the result field is alphameric, the result field contains only numeric characters.
3. If factor 2 is alphameric and the result field is numeric, a minus zone is moved into the rightmost position of the result field if the zone from the rightmost position of factor 2 is a D (minus zone). However, if the zone from the rightmost position of factor 2 is not a D, a positive zone is moved into the rightmost position of the result field. Other result field positions contain only numeric characters.
4. If factor 2 and the result field are alphameric, only the number of characters needed to fill the result field are moved.

If factor 2 is shorter than the result field and:

1. If factor 2 is either numeric or alphameric and the result field is numeric, the digit portion of factor 2 replaces the contents of the leftmost positions of the result field. The sign in the rightmost position of the result field is not changed.
2. If factor 2 is either numeric or alphameric and the result field is alphameric, the characters in factor 2 replace the equal number of leftmost positions in the result field. No change is made in the zone of the rightmost position of the result field.

MSG (Display a Diagnostic Message)

Conditioning Indicators	**Factor 1**	**Operation**	**Table or Message Text**	**Resulting Indicators**
Optional	*Blank*	MSG	*Required*	*Optional*

Factor 1: Blank
Factor 2: MIC or message text

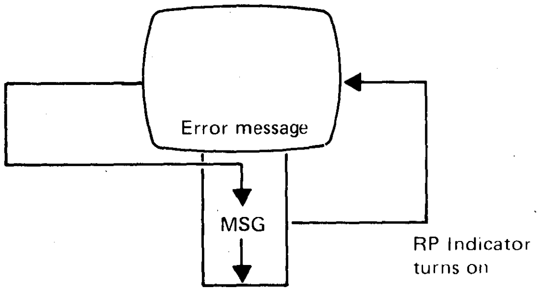
Interrupts the processing of a display and causes the current display to be shown with a user message on the bottom line of the display.

The message overlays any data that is on the bottom line of the display. Output fields conditioned by indicators that are on are updated. The cursor, unless placed again by means of an indicator on a D-specification, remains where it was when the MSG operation occurred. Fields that were protected remain protected; fields that were unprotected remain unprotected, even if the indicator that controls these attributes changes. Data that the operator has entered is not cleared.

If a MSG operation is performed in processing that has no associated display (ES processing, for example), the message is shown on the bottom line of the current display. The operator cannot type data into any field on the display. If an MSG operation is run while the WSU display is shown, the display is cleared before the message is displayed.

The processing of C-specifications stops, and the operator must press the Error Reset key before continuing. The operator, after pressing the Error Reset key, can enter data into any unprotected input field. When the operator responds by pressing the Enter key or a user command key, processing resumes with the first operation for the display. The RP indicator turns on and can be used to condition operations that should or should not repeat. The message is cleared when the Error Reset key is pressed.

The MSG operation performs in the same way, whether or not the display has been prepared for processing. The display is shown with the message, the operator responds, and then the display's processing begins again.



You can code conditioning indicators in columns 9 through 17 and resulting indicators in columns 54 through 59. The indicators in columns 54 through 59 turn on when the MSG operation runs. You can use these indicators to condition attributes of the display (for example, sound the alarm) or of certain fields (for example, blink these fields).

Columns 33 through 52 can contain message text or an MIC. Message text must be enclosed in apostrophes. Code two consecutive apostrophes to represent one apostrophe in text.

An MIC is a 4-digit number that references the appropriate message in your message member. Code the MIC in columns 33 through 36. Columns 23 through 30 of the J-specification must contain the name of your message member.

You can continue text to columns 7 through 52 of the following C-specification line by coding any character except a blank in column 53 of the line you want to continue and coding the remainder of the text (left-adjusted) in columns 7 through 52 of the next line. Column 7 of the continuation line must not be an asterisk (*). WSU allows 64 characters of message text.

MULT (Multiply)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Required*	MULT	*Required*	*Required*	*Optional*

Factor 1: Numeric field or literal
Factor 2: Numeric field or literal

Multiplies factor 1 by factor 2 and places the product in the result field.

Factor 1 and factor 2 do not change unless either is also the result field. The result field should be large enough to hold the product; the length should be specified in columns 49 through 51 (result field length). The maximum product length equals the length of factor 1 plus the length of factor 2.

MVR (Move Remainder)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	MVR	*Blank*	*Required*	*Optional*

Factor 1: Blank
Factor 2: Blank

Moves the remainder from the previous divide operation to the result field.

The MVR operation must immediately follow the divide operation.

The maximum length of the remainder (including decimal positions) is 15. The number of significant decimal positions is the greater of:

- The number of decimal positions in factor 1 of the previous divide operation
- The sum of the decimal positions in factor 2 and the result field of the previous divide operation.

The maximum whole number positions in the remainder is equal to the whole number positions in factor 2 of the previous divide operation.

You cannot code the MVR operation after a divide operation that you half-adjusted (column 53).

PRTY (Priority)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Blank	*Blank*	PRTY	*Required*	*Blank*	*Blank*

Factor 1: Blank
Factor 2: 0, 1, 2, or 3

Assigns the expected frequency of use to a subroutine.

When specified, a PRTY operation must immediately follow a BEGSR operation. Factor 2 must contain 0, 1, 2, or 3. The highest priority is 3, the lowest priority is 0. If the PRTY operation is not specified in a subroutine, WSU assigns priority 0 to that subroutine.

Refer to *Assigning Priority to Display Formats and Subroutines* in Appendix B for an explanation of the use of the PRTY operation.

PUT (Write a Transaction File or Master File Record)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	PUT	*Required*	*Blank*	*Optional*

Factor 1: Blank
Factor 2: Transaction file or master file name

Writes one record to the transaction file or master file.

A PUT operation is required in a program only if records are added, updated, or inserted.

For a PUT operation to a master file, columns 54 and 55 must be blank. For a PUT operation to the transaction file, columns 54 and 55 can indicate the type of record to be written or can be left blank. You can code conditioning indicators in columns 9 through 17; however, columns 43 through 53 and columns 56 through 59 must be blank.

If the PUT operation does not specify a record type, WSU refers to the most recent GET operation that occurred for the file since the most recent mode change and assumes that the record type recognized and saved from this operation is the record type for the PUT operation.

PUTN (Add to a Master File)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	PUTN	*Required*	*Blank*	*Required*

Factor 1: Blank
Factor 2: Master file name

Adds one record to a master file.

Factor 2 must be the name of an indexed master file. You can code conditioning indicators in columns 9 through 17; however, columns 43 through 53 and columns 56 through 69 must be blank. Columns 54 and 55 must contain the type of the record to be added.

When a PUTN operation runs, WSU pulls out fields of the indicated record type from the field area and adds the record to the file.

The key of the added record is the contents of the key field (indicated on the F-specification) at the time the PUTN operation runs. The contents of the GET field(s) specified on the M-specification do *not* indicate the key of the added record. The record key must not be hex FFs.

PUTS (Show a Display)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	PUTS	*Required*	*Blank*	*Blank*

Factor 1: Blank
Factor 2: Display format name

Causes the display named in factor 2 to be selected for processing.

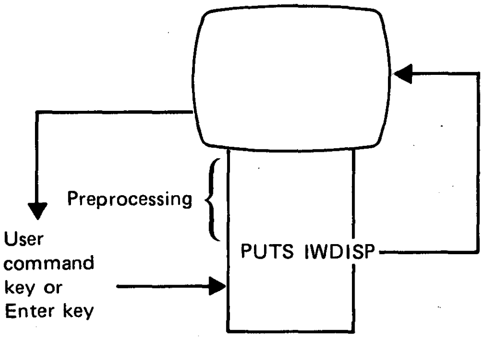
Columns 43 through 59 must be blank. Columns 9 through 17 can contain conditioning indicators.

Factor 2 can be the name of another display in the program or the name of the current display (which causes the display to be shown again).

Factor 2 can name a processing level display only if a PUTS operation occurs in the processing for that preprocessed display.

By using command keys and function keys, the operator can interrupt normal processing to bypass the current display, to view a help display for the program, or to do alternative processing after the PUTS operation.

Processing Level Display named IWDISP



The following chart shows how WSU typically selects the display that follows a display shown by a PUTS operation during an enter mode sequence:

Refer to *Using PUTS in Special Processing Level Calculations* in Chapter 7 for special uses of the PUTS operation.

RANGE (Range Check)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Required*	RANGE	*Required*	*Required*	*Required*

Factor 1: Field or literal
Factor 2: Field or literal

Compares factor 1 with both factor 2 and the result field and, as a result of the comparison, indicators turn on as follows:

High: Factor 1 is greater than the higher value.

Low: Factor 1 is less than the lower value.

Equal: Factor 1 is equal to one of the values or between the two values.

Factor 1, factor 2, and the result field must all be the same type: either alphameric or numeric.

See Chapter 7, *Coding Processing*, for a list of coding rules for literals.

SETOF (Set Off)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	SETOF	*Blank*	*Blank*	*Required*

Factor 1: Blank
Factor 2: Blank

Sets off the indicators named in columns 54 through 59.

Code the indicators that can be set off in columns 54 through 59. The headings for columns 54 through 59 (plus or high, minus or low, zero or equal) have no meaning for this operation. Refer to Chapter 3, *Reserved Fields and Indicators*, for a description of each of the indicators.

Note: Indicators CG, DL, IJ, IN, IW, and RV cannot be set off by the SETOF operation.

SETON (Set On)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	SETON	*Blank*	*Blank*	*Required*

Factor 1: Blank
Factor 2: Blank

Sets on the indicators named in columns 54 through 59.

Code the indicators that can be set on in columns 54 through 59. The heading for columns 54 through 59 (plus or high, minus or low, zero or equal) have no meaning for this operation. Refer to Chapter 3, *Reserved Fields and Indicators*, for a description of each of the indicators.

Note: Indicators AC, CG, DL, IJ, IN, IS, IW, RP, and RV cannot be set on by the SETON operation.

SORTA (Sort the Elements of an Array)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	SORTA	*Required*	*Blank*	*Blank*

Factor 1: Blank
Factor 2: Array name

Sequences the elements of an array named in factor 2.

Factor 1, the result field, half-adjust, and resulting indicators must be blank if a SORTA operation is specified. Indexing is not allowed for factor 2.

You can ensure that the elements of the array are in the proper sequence for a LOKUP operation by performing a SORTA operation. The array specified in factor 2 is sorted into the sequence specified in the E-specification for the array. If no sequence is specified, the array is sorted into ascending sequence. The standard EBCDIC collating sequence is used for the SORTA operation.

SUB (Subtract)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Required*	SUB	*Required*	*Required*	*Optional*

Factor 1: Numeric field or literal
Factor 2: Numeric field or literal

Subtracts factor 2 from factor 1 and places the difference in the result field.

Factor 1 and factor 2 do not change, unless either is also the result field.

Subtracting a field from itself is a method of setting the result field to zero.

TAG

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Blank	*Required*	TAG	*Blank*	*Blank*	*Blank*

Factor 1: Label
Factor 2: Blank

Names the statement that a GOTO operation can branch to.

Factor 1 must begin in column 18 with an alphabetic character and must be 1 to 6 characters long.

Factor 1 cannot be the same as:

- Factor 1 of a BEGSR operation
- Factor 1 of an ENDSR operation
- Any other field name defined in the program.

The TAG operation does not use factor 2 and the result field, and does not allow conditioning indicators in columns 9 through 17.

Refer to *Operations* in Chapter 7 for examples of coding the GOTO and TAG operations.

TIME (Get the Time of Day and Date)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	TIME	*Blank*	*Required*	*Blank*

Factor 1: Blank
Factor 2: Blank

Moves the system time-of-day or the system time-of-day and the system date into the result field.

If the result field specifies a 6-digit numeric field with zero decimal positions, the time-of-day is placed in the field. If the result field specifies a 12-digit numeric field with zero decimal positions, the time-of-day is placed in the first six positions and the system date is placed in the last six positions.

Entries in factor 1 (columns 18 through 27), factor 2 (columns 33 through 42), half-adjust (column 53), and resulting indicators (columns 54 through 59) are not allowed.

XFOOT (Sum the Values of an Array)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	XFOOT	*Required*	*Required*	*Optional*

Factor 1: Blank
Factor 2: Array name

Adds the elements of factor 2 and places the sum in the result field. Indexing is not allowed in factor 2.

For information on how to code arrays and array operations, refer to Chapter 5, *Coding Arrays*, and Chapter 14, *E-(Array) Specifications*.

Z-ADD (Zero and Add)

Conditioning Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators
Optional	*Blank*	Z-ADD	*Required*	*Required*	*Optional*

Factor 1: Blank
Factor 2: Numeric field or literal

Adds factor 2 to a field of zeros (result field) and places the sum in the result field.

Factor 2 does not change, unless it is also the result field.

Z-SUB (Zero and Subtract)

Conditioning Indicators	**Factor 1**	**Operation**	**Factor 2**	**Result Field Name**	**Resulting Indicators**
Optional	*Blank*	Z-SUB	*Required*	*Required*	*Optional*

Factor 1: Blank
Factor 2: Numeric field or literal

Subtracts factor 2 from a field of zeros (result field) and places the difference in the result field.

Factor 2 does not change, unless it is also the result field.

Chapter 19. File Processing Procedures

This chapter includes detailed information for using the WSU procedures. The major functions, prompt display, command statement format, parameters, and examples are included for each of the three WSU procedures discussed in Chapter 4, *Coding Files*.

WSU EXTRACT PROCEDURE (WSUTXEX)

Major Functions

The WSU Extract procedure copies records from a WSU transaction file or a non-WSU file (but not an immediate access file). The difference between these two types of files is that the WSU file has control records, data records, and 13 bytes of trailer information in each record. The non-WSU file may or may not have control records and trailers, but the file is not usable by a WSU program.

The WSU Extract procedure can specify that one or all work session chains are extracted from (pulled out of) a WSU file in logical order, that one or all work session chains are extracted from a WSU file in physical order, or that all data-containing records are extracted.

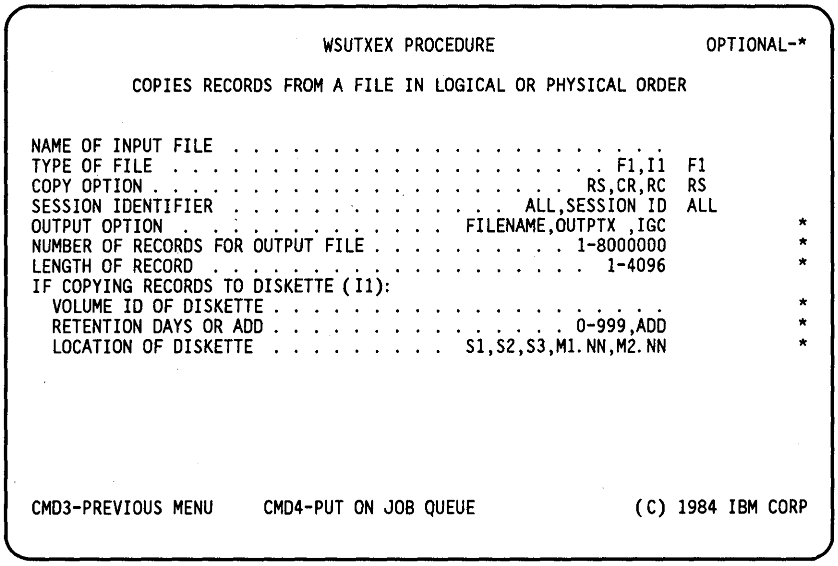
The extracted records can be printed, displayed, put in a disk file, put in a new diskette file, or added to an existing diskette file.

If the extracted records are put in a disk or diskette file, the output record length can be shortened to exclude trailer information. Also, the number of records allocated for the output file can be reduced to allow for leaving out control records.

Notes:

1. The WSU Extract procedure can be used to do the functions provided by SUBR22, a subroutine of RPG.
2. If logically deleted (not chained) records exist in the file, the chain should be pulled out in logical order because records which are not removed from the chain cannot be identified and cannot otherwise be excluded.
3. The transaction file is created as a direct file; after it is copied it becomes a sequential file.

Prompt Display



```
                          WSUTXEX PROCEDURE                   OPTIONAL-*

          COPIES RECORDS FROM A FILE IN LOGICAL OR PHYSICAL ORDER

NAME OF INPUT FILE . . . . . . . . . . . . . . . . . . . . . . .
TYPE OF FILE . . . . . . . . . . . . . . . . . . . . . . F1,I1  F1
COPY OPTION . . . . . . . . . . . . . . . . . . . . . RS,CR,RC  RS
SESSION IDENTIFIER . . . . . . . . . . . . . . . ALL,SESSION ID  ALL
OUTPUT OPTION . . . . . . . . . . . . . . FILENAME,OUTPTX ,IGC              *
NUMBER OF RECORDS FOR OUTPUT FILE . . . . . . . . . . . 1-8000000              *
LENGTH OF RECORD . . . . . . . . . . . . . . . . . . . . . 1-4096              *
IF COPYING RECORDS TO DISKETTE (I1):
  VOLUME ID OF DISKETTE . . . . . . . . . . . . . . . . . . . . .              *
  RETENTION DAYS OR ADD . . . . . . . . . . . . . . . . 0-999,ADD              *
  LOCATION OF DISKETTE . . . . . . . . . . S1,S2,S3,M1.NN,M2.NN              *




CMD3-PREVIOUS MENU     CMD4-PUT ON JOB QUEUE               (C) 1984 IBM CORP
```

Parameters

Name of input file: Specifies the name of the input file.

Type of file (disk F1, diskette = I1):

F1	Specifies that the input file is a disk file. If no parameter is entered, F1 is assumed.
I1	Specifies that the input file is a diskette file. This file must not be on disk when the WSUTXEX command is entered.

Copy option:

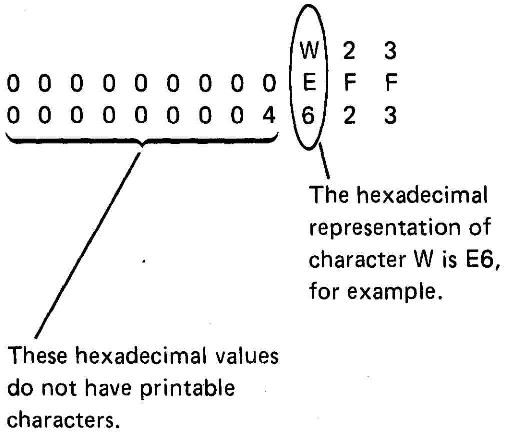
RS	Specifies that header and detail records are to be copied in logical order from a WSU transaction file. Records removed from the chain are not copied. If no parameter is entered, RS is assumed.
RC	Specifies that header and detail records are to be copied in physical order from a WSU transaction file or a non-WSU file.
CR	Specifies that all records that contain at least one nonblank position are to be copied in physical order from a non-WSU file.

Session identifier:

ALL	Specifies that all header and detail records are to be copied.
session ID	Specifies that only those header and detail records that have the specified session ID are to be copied.

Output option:

file name	Specifies the name of the file the WSUTXEX procedure is to create. This file must not exist when the WSUTXEX command is entered. The output file is allocated as a temporary sequential file.
OUTPTX	Specifies that copied records are to be listed on the system list device. The relative record number that a record has in the input file is displayed before that extracted record. The hexadecimal representation of the characters in the records is shown as follows:



```
                           W  2  3
0 0 0 0 0 0 0 0 0  E  F  F
0 0 0 0 0 0 0 0 4  6  2  3
```

The hexadecimal representation of character W is E6, for example.

These hexadecimal values do not have printable characters.

IGC	Specifies that copied records containing ideographic characters (IGC) are to be listed on the current syslist device. If the current syslist device is a display that is not capable of displaying ideographic data, the ideographic characters will be replaced by dots. If the current syslist device is a printer that is not capable of printing ideographic data, the printer will treat the ideographic data as unprintable characters. If the printer stops, enter a 0 to hold the print job, or enter a 1 to replace the ideographic characters with blanks on the printed output.

The number of extracted records is shown at the end of the printout or the display.

Note: If the file name parameter is blank, extracted records are treated as if OUTPUT has been specified. If a printer is the system list device, the hexadecimal representation is shown only for nonprintable characters. The previous example would be printed as follows:

```
         W23
000000000
000000000
```

Number of records for output file: Specifies the number of records that are allocated for the output file. Any number from 1 through 8000000 can be specified. If a number of records is not specified, the number of records actually used in the input file is used.

The number of records must be specified when extracting records from a file whose retention is S or J.

Length of records: Specifies the length of the records for the output file. If this length is longer than the records in the input file, records are extended and padded with blanks when they are copied to the output file. If this length is shorter than the records in the input file, records are truncated to this length when they are copied to the output file. If the record length is not specified, the record length of the input file is assumed.

For example, to remove trailers from a WSU transaction file, the record length specified for the output file should be 13 positions shorter than the input file record length.

If copying records to diskette (I1):

Volume ID of diskette: Specifies the volume ID of a diskette to receive the copied records. If a volume ID is specified, the file name parameter is the name of a new or existing diskette file to which records are copied.

Retention days or ADD:

0-999	Specifies that a new diskette file is to be created and the number of days it is to be retained. Any decimal number from 0 through 999 can be specified (999 indicates a permanent disk). If no retention period is entered, 1 day is assumed.
ADD	Specifies that an existing diskette file is to have the copied records added to it.

Location of diskette:

S1, S2, or S3	Specifies the diskette slot containing the first diskette to be used. If a parameter is not specified, S1 is assumed.
M1.nn or M2.nn	Specifies the magazine location that contains the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. nn is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. (Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01).

Command Statement Format

- To copy records from a file and list them or copy them to a disk file:

```
                           ┌  ┐ ┌  ┐ ┌          ┐
WSUTXEX   input file name, │F1│,│RS│,│ALL       │,
                           │I1│ │RC│ │session id│
                           └  ┘ │CR│ └          ┘
                                └  ┘

          ┌                                                      ┐
          │OUTPTX                                                │
          │                                                      │
          │                  ┌                 ┐ ┌             ┐ │
          │output file name, │number of records│,│record length│ │
          │                  └                 ┘ └             ┘ │
          └                                                      ┘
```

- To copy records from a file and add them or copy them to a diskette file:

```
                           ┌  ┐ ┌  ┐ ┌          ┐
WSUTXEX   input file name, │F1│,│RS│,│ALL       │,output file name,
                           └  ┘ │RC│ │session id│
                                │CR│ └          ┘
                                └  ┘

          ┌                 ┐ ┌             ┐             ┌              ┐
          │number of records│,│record length│,volume id,  │retention days│,
          └                 ┘ └             ┘             │1             │
                                                          │ADD           │
                                                          └              ┘

          ┌     ┐
          │S1   │
          │S2   │
          │S3   │
          │M1.nn│
          │M2.nn│
          └     ┘
```

Examples

- Print all data records in the WSU transaction file TRANF00 for session W0, including those records that the operator cannot review in the W0 chain of records. Print records in physical order and show the hexadecimal representation of only nonprintable characters.

  ```
  WSUTXEX TRANF00,F1,RC,W0
  ```

- Print the chain of data records in WSU transaction file TRANF01 for session W1. Print records in logical order and have the relative record number of a record it has in the input file come before each record. Show the hexadecimal representation of each character whether it is printable or not printable.

  ```
  WSUTXEX TRANF01,F1,RS,W1,OUTPTX
  ```

- Extract all data records in the WSU transaction file TRANF02 for session W2 and copy them to a new diskette file, TRANFD2. The volume ID of the diskette is PRIV97. No more than 30 records are pulled out.

  ```
  WSUTXEX TRANF02,F1,RS,W2,TRANFD2,30,,PRIV97
  ```

Note: Control records are not copied to the diskette file; however the trailer remains in each data record.

- Extract as many as 30 data records in the WSU transaction file TRANF02 for session W3, and copy them to an old diskette file, TRANFD2. The diskette volume ID is PRIV97.

  ```
  WSUTXEX TRANF02,F1,RS,W3,TRANFD2,30,,PRIV97,ADD
  ```

  This example and the previous example show how you can collect work session chains in any order on a diskette. Any chains can be excluded from the diskette. These examples could also be modified to collect work session chains from two or more WSU transaction files or to concatenate a work session chain that is in two or more WSU transaction files.

- Add all nonblank records from a non-WSU direct file, STDDIRC, to an old diskette file, TRANFD2. The diskette volume ID is PRIV97.

  ```
  WSUTXEX STDDIRC,,CR,,TRANFD2,,,PRIV97,ADD
  ```

- Copy records from a WSU transaction file, TRANF018, to a new diskette file, TRANFI18. The diskette is in slot S2 and its volume ID is PRIV97. The file is retained for 7 days. Records are copied in logical order, control records are removed, and trailers (in positions 59 through 71) are removed from the data records.

  ```
  WSUTXEX TRANF018,,RS,ALL,TRANFI18,,58,PRIV97,7,S2
  ```

- This example shows how to copy records from a WSU transaction file on a disk named WSUTRAN and list those records on the system list device.

  ```
  WSUTXEX WSUTRAN
  ```

WSU CREATE PROCEDURE (WSUTXCR)

Major Functions

The WSU Create procedure can be used to create a WSU transaction file from a non-WSU file but not from an immediate access file. An output record length that allows for the addition of a trailer to each record can be specified. A number of records that allows for the addition of control records to the output file can be specified. A session ID can be assigned to all of the output records. Also, a WSU program that processes the newly created file can be specified.

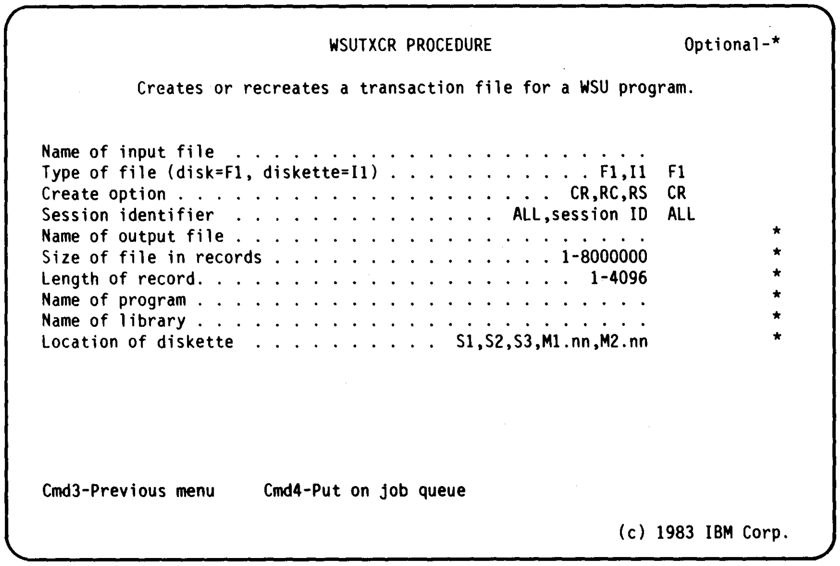
The WSU Create procedure can also be used to create a WSU transaction file from an existing WSU transaction file. This procedure can be used to:

- Remove unrecognized records or to allow another WSU program to use a subset of the file.
- Allow a specified display station to access all the chains as a single chain.
- Allocate additional or fewer records for the file.
- Recover inserted records that were lost when the WSU program ended abnormally. These records can be recovered, but they will not be in the originally intended chain positions. If the session chain has changed since the records were lost, these records might not be able to be recovered.

When creating a WSU transaction file, either originally or again, the input file can be on disk or diskette and the output file is always a direct, disk file.

You can use the WSUTXCR procedure to recover or remove any records that are logically deleted or partially inserted into a file.

Prompt Display



```
                           WSUTXCR PROCEDURE                   Optional-*

          Creates or recreates a transaction file for a WSU program.


Name of input file . . . . . . . . . . . . . . . . . . . . . . .
Type of file (disk=F1, diskette=I1) . . . . . . . . . . . F1,I1  F1
Create option . . . . . . . . . . . . . . . . . . . . . CR,RC,RS  CR
Session identifier . . . . . . . . . . . . . . . ALL,session ID  ALL
Name of output file . . . . . . . . . . . . . . . . . . . . . .        *
Size of file in records . . . . . . . . . . . . . . . . 1-8000000       *
Length of record. . . . . . . . . . . . . . . . . . . . . 1-4096       *
Name of program . . . . . . . . . . . . . . . . . . . . . . . .        *
Name of library . . . . . . . . . . . . . . . . . . . . . . . .        *
Location of diskette . . . . . . . . . . S1,S2,S3,M1.nn,M2.nn          *





Cmd3-Previous menu     Cmd4-Put on job queue

                                                       (c) 1983 IBM Corp.
```

Parameters

Name of input file Specifies the name of the input file.

Type of file (disk F1, diskette = I1):

F1	Specifies that the input file is on disk. If no parameter is entered, F1 is assumed.
I1	Specifies that the input file is on diskette. The input file must not exist on disk when the WSUTXCR command is entered.

Create option:

CR	Specifies that a WSU transaction file is to be created from records from the input file. All the records that contain at least one nonblank position are copied in physical order from the input file to the output file. Blank records are added to the file in the relative record positions the control records will occupy later. If no parameter is entered, CR is assumed.
RC	Specifies that a WSU transaction file is to be *recreated* from the input file which is a WSU file. All control, header, and detail records are copied in physical order from the input file to the output file.
RS	Specifies that a WSU transaction file is *created* from a WSU transaction file. All header records are copied in logical order from the input file to the output file.

Session identifier:

ALL	Indicates that the session ID in each input file record is to be copied unchanged to the output file. The ID is assumed to be in the next-to-last 2 bytes in the input record. If no parameter is specified, ALL is assumed.
session ID	Specifies the session ID (W1 W2), that is to be placed in each output record.

Name of output file: Specifies the name of the file the WSUTXCR procedure is to create. This file must not exist when the WSUTXCR command is entered.

Size of file in records: Specifies the number of records that are to be allocated for the output file. If the number of records is not specified, the number of records actually used in the input file is used. Any number of records from 1 through 8000000 can be specified. When a WSU transaction file is created from a non-WSU file, the output file requires more records than the input file to allow storage for a job control record, a work session control record for each record chain, and a blank record at the logical end of each record chain.

Length of record: Specifies the length of the records for the output file. If CR (create WSU transaction file) is specified, the record length parameter should allow for the 13-byte trailer at the end of each record in the output file. If the record length parameter is omitted, the default length used is the input record length, but will not include the required 13 bytes for the trailer.

Name of Program: Specifies the name of a generated WSU program that is to use the created WSU transaction file. Information in this WSU program is used to identify header records and detail data records in the file and logically chain them together. The named WSU program should not contain any set of tests for identifying a record type in which the tests could be satisfied by a record containing blanks in the user data portion of the record.

WSU can be specified as the program name. This would be done if the WSU program for which the transaction file is being created is not available (for example, it has not yet been generated) or if blank records in the file would be mistakenly identified as data records for a defined record type. If WSU is specified as the program name, all nonblank detail records in the output file are identified as detail data records.

If the program name is omitted, a trailer must be present in each input record because the last byte of the trailer is used to identify header records and detail records. A header record contains hex F7 in the last byte of its trailer. A detail record contains hex F6 in the last byte of its trailer.

Name of library: Specifies the name of the library that contains the generated WSU program. The library name parameter can be specified only if the program name is specified. If a library name is omitted when a program name is specified, the default library is the current library. The library name is ignored if WSU is specified as the program name.

Diskette location:

S1, S2, or S3	Specifies the diskette slot containing the first diskette to be used for input. If a parameter is not specified, S1 is assumed.
M1.nn or M2.nn	Specifies the magazine location that contains the first diskette to be used. M1 indicates the first magazine and M2 indicates the second magazine. nn is a decimal value from 01 through 10 that identifies the location of the diskette in magazine. (Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01).

Command Statement Format

```
WSUTXCR  input file name,┌  ┐ ┌  ┐ ┌            ┐,output file name,
                         │F1│,│CR│,│ALL         │
                         │I1│ │RC│ │session id  │
                         └  ┘ └  ┘ └            ┘

         ┌                 ┐ ┌             ┐ ┌            ┐
         │number of records│,│record length│,│WSU         │,
         └                 ┘ └             ┘ │program name│
                                             └            ┘

         ┌                ┐ ┌     ┐
         │library name    │,│S1   │
         │current library │ │S2   │
         └                ┘ │S3   │
                            │M1.nn│
                            │M2.nn│
                            └     ┘
```

Examples

- Create a WSU transaction file, TRANF02, from a non-WSU diskette file, TRANFD2. Extend the output record length to allow room for trailers in positions 59 through 71. Increase the number of records to 5000 when the output file is created. A WSU program named WSUFT02 processes the output file.

  ```
  WSUTXCR TRANFD2,I1,CR,ALL,TRANF02,5000,71,WSUFT02
  ```

  Because ALL is specified, each input file record is assumed to have a session ID in the next-to-last 2 bytes in each record. This ID is preserved when records are copied.

- Create a WSU transaction file, TRANF018, from a non-WSU diskette file, TRANFI18. Extend the output record length to allow room for trailers in positions 59 through 71. Assign all records to session W0, allocate 1000 records for the output file, and allow WSU program WSUFT018 to process the file.

  ```
  WSUTXCR TRANFI18,I1,CR,W0,TRANF018,1000,71,WSUFT018
  ```

- Create a WSU transaction file so that another WSU program and a specified display station can use it. The WSU file to be duplicated is TRANF01. This file was previously pulled out using the WSUTXEX procedure to make the physical and logical order of the records in the file the same. The new file name is TRANF018, and display station W1 and WSU program WSUFT018 in library WSUTEST are able to use it.

  ```
  WSUTXCR TRANF01,,RC,W1,TRANF018,,,WSUFT018,WSUTEST
  ```

  If this example were modified to use a second set of added trailers, a new TRANF01 file could be duplicated for the original sessions from any updated TRANF018 records.

- Create a WSU transaction file in order to recover inserted records that cannot be reclaimed because the session was used after the WSU program ended abnormally. The file to be duplicated is TRANF018 and the new WSU file is TRANFR18.

  ```
  WSUTXCR TRANF018,,RC,ALL,TRANFR18
  ```

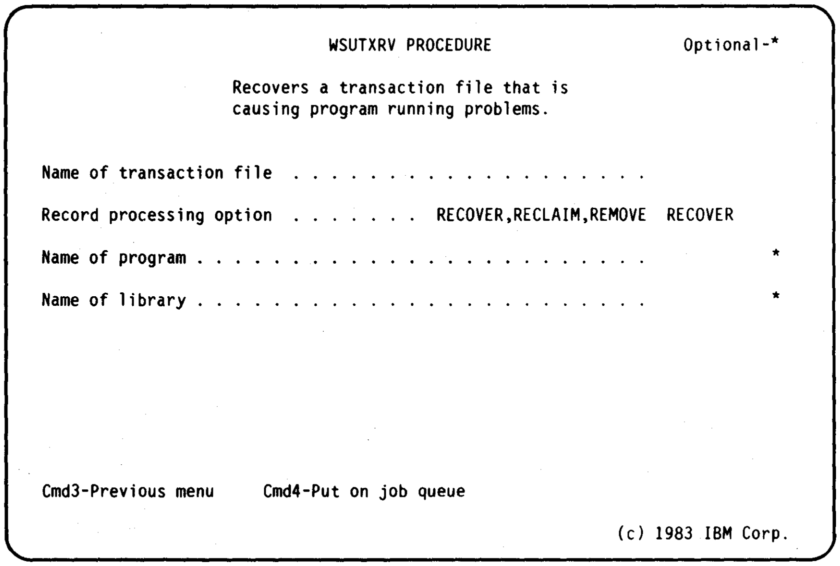
WSU RECOVER PROCEDURE (WSUTXRV)

Major Function

The WSU Recover procedure recovers a transaction file that has problems when a WSU program is running. Unidentified records are removed from the file, and blanks are placed in the locations that they occupied. (Unidentified records are those records that do not match any of the record types that the WSU program recognizes.)

In addition to recovering a file, the WSU Recover procedure can be used to reclaim or remove partially inserted or logically deleted records. These records can be in the file after the WSU program or a WSU session ends abnormally during insert mode.

Prompt Display



```
                          WSUTXRV PROCEDURE                    Optional-*

                     Recovers a transaction file that is
                     causing program running problems.

Name of transaction file  . . . . . . . . . . . . . . . . . . .

Record processing option  . . . . . . .  RECOVER,RECLAIM,REMOVE  RECOVER

Name of program . . . . . . . . . . . . . . . . . . . . . . .              *

Name of library . . . . . . . . . . . . . . . . . . . . . . .              *





Cmd3-Previous menu     Cmd4-Put on job queue

                                                        (c) 1983 IBM Corp.
```

Parameters

Name of transaction file: Specifies the name of the file to be recovered.

Record processing option:

RECOVER	Specifies that control records are to be restored so there is no flag to indicate that records have been deleted.
RECLAIM	Specifies that partially inserted records are to be reclaimed, that is, operators will be able to use the records. Operators are able to view records after the records have been reclaimed.
REMOVE	Specifies that partially inserted and deleted records are removed from the transaction file.

Note: If neither RECLAIM nor REMOVE is specified, any partially inserted or deleted records remain partially inserted or logically deleted in the transaction file.

Name of program: Specifies the name of a generated WSU program that contains the record type format that is to be used to determine which records are correct or not correct. Records that do not match an allowed record type are removed from the transaction file. If the program name is omitted, this parameter and the library name parameter are not used. If the program name is omitted but the library name parameter is specified, the program name is prompted for by the system.

Name of library: Specifies the name of the library containing the WSU program. The library name can be specified only if the program name is specified. If the library name is omitted when the program name is specified, the current library is assumed.

Command Statement Format

```
WSUTXRV  file name,┌RECOVER┐,┌program name┐,┌library name   ┐
                   │RECLAIM│ └            ┘ │current library│
                   │REMOVE │                └               ┘
                   └       ┘
```

Examples

- Reclaim all partially inserted records in a transaction file named TRANF02.

  ```
  WSUTXRV  TRANF02,RECLAIM
  ```

- Delete records from a transaction file named TRANF03 that do not match the record types described in a WSU program named PGM03. The WSU program is in a user library named LIB03.

  ```
  WSUTXRV  TRANF03,,PGM03,LIB03
  ```

- Recover a transaction file named TRANF04 and remove partially-inserted and logically-deleted records from it. Refer to *Appendix B, Additional Topics and Programming Considerations* for a discussion of partially-inserted and logically-deleted records.

  ```
  WSUTXRV  TRANF04,REMOVE
  ```

Chapter 20. Ideographic Considerations

Not every WSU programmer uses ideographic characters. This chapter is for those who use ideographic characters (IGC) and alphameric-katakana characters (A/N/K).

This chapter discusses the uses, requirements, and restrictions you should consider when using ideographic, alphameric, and Katakana characters.

Using Ideographic Characters

To use ideographic characters in a WSU program, you must:

- Have the ideographic feature on your system.
- Sign on to an ideographic-capable display by entering Y (yes) to the IGC session prompt on the sign on screen.
- Enter Y (yes) in column 44 of the J-specification for your WSU program.

Space Requirements

Alphameric characters are one position each. Ideographic characters are two positions each. An ideographic data string starts with a shift-out character (SO) of one position, followed by ideographic characters of two positions each, followed by a shift-in character (SI) of one position. The shift-out character is hexadecimal (0E), and the shift-in character is hexadecimal (0F).

The following figure shows a two-character ideographic data string:

An ideographic data string always is:

- An even number of positions.
- Four or more positions long (the shift-out character, ideographic character, and the shift-in character)
- Bracketed by a shift-out and shift-in characters.

Usage Restrictions

Alphameric, Katakana, and ideographic characters can be used for input, output, constant, and literal data. Ideographic characters cannot be used for file names, program names, or library names.

Display fields can be created, extended, or moved so that the field is on two or more lines of the display. If such a field contains ideographic data, the last ideographic character on each line must be followed by a shift-in character, and the first ideographic character on the next line must be preceded by a shift-out character.

If an ideographic constant is coded on the D-specification so that the constant continues onto the next line of the D-specification, the first coded line must have a shift-in character in column 78 or 79, and column 7 of the next line of the D-specification must contain a shift-out character.

WSU will remove the shift-in character coded in column 78 of the D-specification, the blank in column 79, and the shift-out character in column 7 of the continuation line. This feature is called concatenation. The rules of concatenation are given in *Ideographic Coding* later in this chapter.

An ideographic-capable display station can display uppercase alphabetic characters. An ideographic keyboard can enter only the uppercase alphabetic characters.

WSU programs treat ideographic data as alphameric data. Illogical processing and edit codes will pass the syntax check. For example, searching ideographic data for a high/low condition may not give a meaningful result.

You must check the output of your WSU programs to ensure that you are using the ideographic data correctly.

Ideographic Coding

J-specification	For the details of coding the J-specification for Ideographic Mode, see Chapter 13, column 44.
D-specification	For the details of coding the D-specification for ideographic and Katakana data, see Chapter 16, D-specification, column 27.

Concatenation Rules

Concatenating ideographic characters on a D-specification follows these rules:

- If a SI is in column 78 of a D-specification; column 79 is ignored.
- If a SI is in column 78 of a continued D-specification line and the following line has a SO in column 7, any entry in column 79 of the first line is ignored, the SI and SO characters are dropped, and the constant is displayed without gaps between the ideographic data.
- If a SI is in column 79 of a continued D-specification and the following line has a SO in column 7, the SI and SO are dropped, and the constant is displayed without gaps in the ideographic data.
- A SO in column 79 of a D-specification is ignored.
- A SO in column 78 and a SI in column 79 of a D-specification sheet are ignored.

Sorting	For the details of sorting ideographic characters, see *IBM S/36 Ideographic Sort Guide,* SC09-1054.

Appendix A. Using Self-Check Fields

Self-checking numbers offer some amount of protection against clerical errors and fraud. Self-checking is particularly useful in applications that use account numbers.

The System/36 offers two methods of self-checking: modulus 10 and modulus 11. If a self-checking method (T for modulus 10 or E for modulus 11) is specified in column 30 of the D-specification for an input field, System/36 determines a self-check digit for the field's contents, using the specified self-check method. That self-check digit is compared to the rightmost position of the input field. (Nulls and blanks are considered to be self-check digits of zero.) If the self-check digit matches the rightmost position of the input field, the contents of the input field are allowed and the operator can continue. If the numbers do not match, the contents of the input field are not allowed, and a keyboard error is displayed. The operator must then enter an allowed number before continuing. For example, the number 123216 is a valid modulus 10 number. The self-check digit is 6 and must be in the rightmost position of the input field. The remaining digits, 12321, are used to determine whether the self-check is successful.

The following pages describe the methods for determining the self-check digit for the contents of an input field. After reviewing how the self-check digit is determined, you might want to write a program that generates input numbers that successfully complete a self-check. You can, for example, use the generated numbers as a basis for assigning account numbers, item numbers, or security codes.

Modulus 10

To determine the modulus 10 self-check digit, do the following:

1. Disregard the rightmost digit. This digit is not used in the remaining steps, but is used in determining whether the self-check is successful.
2. Multiply the units position (the rightmost digit of what resulted from step 1) and every alternate position of that number by 2.
3. Add the digits in the products to the digits in the numbers that were not multiplied in step 2 (again, excluding the original rightmost digit).
4. Subtract the sum from the next higher number ending in zero (0).

The difference is the self-check digit. Compare this digit with the rightmost digit of the input field. If those digits are the same, the self-check is successful.

For example, suppose you specify modulus 10 self-checking for an input field and the operator enters 123216:

Number to be self-checked	1 2 3 2 1
New rightmost position and every alternate position	1 3 1
Multiply by 2	2 6 2
Digits not multiplied	2 2
Add	2+2+6+2+2 = 14
Next higher number ending in 0	20
Subtract	20 - 14 = 6
Self-check digit	**6**

The self-check digit matches the rightmost digit of the entered number (in this case, 6). The self-check is successful.

Note: Remember that the self-check digit is always in the rightmost position of the input field.

Modulus 11

To determine the modulus 11 self-check digit, exclude the rightmost character and perform the following calculation on the remaining digits.

1. Assign a weighting factor to each digit of the entered number. These factors are: 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, 2, 3, and so on, starting with the new rightmost position of the number and progressing toward the high-order digit (the leftmost digit). For example, the input number 991246351 would be assigned the weighting factors as follows:

Number to be self-checked	9 9 1 2 4 6 3 5
Weighting factor	3 2 7 6 5 4 3 2

2. Multiply each digit by its weighting factor.
3. Add the products.
4. Divide this sum by 11.
5. Subtract the remainder from 11.

The difference is the self-check digit. Compare this digit with the rightmost digit of the input field. If those digits are the same, the self-check is successful.

Note: If the remainder from step 4 is 0, the self-check digit is 0. If the remainder is 1, the entered number does not have a self-check digit; you must ensure that numbers with remainders of 1 in step 4 are not used in the fields you define as self-check fields.

For example, suppose you specify modulus 11 self-checking for an input field and the operator enters 123218:

Number to be self-checked	1 2 3 2 1
Weighting factors	6 5 4 3 2
Multiply	6 10 12 6 2
Add	6+10+12+6+2 = 36
Divide	36/11 = 3 plus a remainder of 3
Subtract	11 - 3 = 8
Self-check digit	**8**

The self-check digit matches the rightmost digit of the input number (in this case, 8). The self-check is successful.

Note: Remember that the self-check digit is always in the rightmost position of the input field.

Appendix B. Additional Topics and Programming Considerations

This appendix provides additional information about:

- Assigning priority to display format processing and subroutines
- Saving total fields
- Interactive Communications Feature (ICF)
- Distributed Data Management (DDM)
- Delete-capable file considerations
- Extend file considerations
- Logical delete considerations
- User-defined help displays.

Assigning Priority to Display Format Processing and Subroutines

When a WSU program runs, the entire program may not fit in the assigned main storage. WSU uses a technique of overlaying various segments of a program upon one another. Generally, better program performance occurs if segments of C-specifications that are required least often are overlaid most frequently.

When the C-specifications for a display format or subroutine run, WSU determines whether or not the processing segment for that format is in main storage. If the segment is not in main storage, WSU reads the segment from disk. To make room in main storage, WSU overlays one or more program segments. Segments with the least probability for use in later processing are the best segments to overlay.

The priority of a display format or subroutine allows you to control which processing segments are overlaid. Assign a high priority to a display format or subroutine that is used frequently, which makes the associated segment less likely to be overlaid. The processing segment remains in main storage for more time. Assign a low priority to display formats or subroutines that appear infrequently, which makes the associated processing segments more likely to give up their main storage to more frequently used segments.

Assign display formats priorities in column 46 of an S-specification. An entry of 3 represents the highest priority. Other priority entries, from high to low, are 2, 1, and 0, or blank.

Examples of Assigning Priorities

A sample application has nine display formats, F1 through F9. The predicted relative frequencies for these displays are:

Display	Relative Frequency
F1	6%
F2	6%
F3	67%
F4	10%
F5	2%
F6	2%
F7	2%
F8	3%
F9	1%

Based on these predictions, the assigned priorities are:

Display	Priority
F3	3
F1, F2, F4	2
F5, F6, F7, F8	1
F9	0 or blank

Saving Total Fields

When operators resume or recover a WSU program that has a transaction file, internal totals accumulated during previous use of the program are not available. In order to make them available, you should save them from time to time by writing them to a master file.

To use a master file to keep the totals, create this file with only one record. The fields in the record should be the totals that are to be saved. When it is time to save the totals, issue a PUT operation to the master file.

To restore the totals when operators resume or recover, issue a GET operation conditioned by the RC indicator or RS indicator for the master file during IJ processing.

Interactive Communications Feature (SSP-ICF)

If an SSP-ICF session attempts to start a WSU program or attach to a WSU program, abnormal program ending occurs. A WSU work session is not started for that SSP-ICF session.

Distributed Data Management (DDM)

DDM is an SSP feature that lets WSU use master and transaction files residing on a remote system. When a remote master or transaction file is specified, DDM locates the file by its entry in the Network Resource Directory (NRD). The NRD is created and maintained by the EDITNRD utility provided with DDM. While DDM allows WSU programs to run on remote files, WSU Extract and Create procedures, WSUTXEX and WSUTXCR, should be run on local data files only.

Note: DDM requires SSP-ICF.

Delete-Capable File Considerations

WSU does not allow records to be physically deleted. (An operator can use the Delete command key to logically delete a record; see *Logical Delete Considerations*, below, for more information.)

Therefore, a WSU program ends abnormally if it tries to use a delete-capable transaction file.

A WSU program can, however, use a delete-capable master file. When the program gets a deleted record, a not-found indication occurs. These deleted records cannot be updated.

Also, when a WSU program writes a record that has hex FF in its first position to a delete-capable master file, the session ends abnormally.

Note: A delete-capable file is a file that can contain logically deleted records. The file is not physically compressed for record deletions. In order to support delete-capable files, a System/36 must have extended data management, an optional SSP feature that is specified during system configuration.

Extend File Considerations

WSU can only extend an old file (one specified with DISP-OLD on the FILE statement). If you use the extend function, the file must already exist before the WSU program is run.

Logical Delete Considerations

During review mode, an operator can logically delete transaction file records that were entered for the work session. Logically deleted records are removed from the work session's chain of records.

You can prevent the operator from deleting records by:

- Disabling the Delete command key on the appropriate S-specification.
- Using the MSG or IMSG operation to display a message to the operator if he uses the Delete command key (DL indicator is on). This message should instruct the operator to request some other function. This message does not prevent the record from being deleted (it does not set the DL indicator to off) but it can prevent the function from ending.

If records have been logically deleted from a transaction file, the records are unchained and should be removed from the file before the file is used by a non-WSU program. The logically deleted records can be removed by using the WSU Recover procedure. If you want to maintain a record of the logically deleted records, you can move a code into the record when it is deleted, then use the WSU Extract procedure to remove the deleted records from the transaction file. The removed records can be:

- Printed
- Displayed
- Put in a disk file
- Put in a new diskette file
- Added to an existing diskette file.

Chapter 19 contains more information on using the WSU Recover and WSU Extract procedures.

The WSU Recover procedure or the WSU Extract procedure can be used to remove the unchained records. The existence of unchained records in a transaction file affects the performance of relative record operations requested from the WSU display or during the operation of SUBR22.

User-Defined Help Displays

You can define special help displays for the operator to use during a work session. A help display is an information display that the operator can request from the WSU display in enter mode, review mode, delete mode, or insert mode. You can use help displays to provide additional information about such things as:

- User-defined command key usage
- Display IDs that can be selected
- Program operations and functions.

Help displays are defined on the S-specifications as nonsequenced displays and must follow these rules:

- A format ID (columns 15 and 16) must be specified. The format ID must be unique and cannot be IJ, IW, EW, EJ, or ES.
- Return input fields (Y in column 22) must not be specified.
- Display sequence (Y in columns 41 and 42) must not be specified.
- Review mode record identifying indicators (columns 48 through 53) must not be specified.
- Insert mode record identifying indicators (columns 54 through 59) must not be specified.
- Processing must not be specified (C-specifications).

The operator can respond to the help display by pressing the Enter key, Command Key 1, or any other command or function key that is enabled for the help display. If the operator presses the Enter key, the user display that was being processed before the help display was displayed is resumed. If the operator presses Command Key 1, the WSU display is shown. If the operator presses a command key enabled for the help display, the appropriate action of that command key is performed. User-defined command keys cannot be used to respond to a help display.

Glossary

abnormal termination. A system failure or operator action that causes a job to end unsuccessfully.

AC indicator. See *accept-command-key indicator.*

accept-command-key indicator. Coded as AC. An indicator that signals the status of any current user-defined command key request.

accept-sequence-error indicator. Coded as AE. An indicator that allows operators to bypass required displays.

access method. The way that records in files are referred to by the system. The reference can be consecutive (records are referred to one after another in the order in which they appear in the file), or it can be random (the individual records can be referred to in any order).

AE indicator. See *accept-sequence-error indicator.*

align. To bring into or be in line with another or with others. For example, to align numbers on the decimal point.

allocate. To assign a resource, such as a disk file or a diskette file, to perform a specific task.

alphabetic character. Any one of the letters A through Z (uppercase and lowercase). Some program products extend the alphabet to include the special characters #, $, and @.

alphameric. Consisting of letters, numbers, and often, other symbols, such as punctuation marks and mathematical symbols.

alphameric/Katakana (A, N/K). A combination of alphameric characters and Katakana characters.

alphanumeric. See *alphameric.*

alternative index. An index that is built after an indexed file is created and that provides a different order for reading or writing records in the file.

alternative indexed file. A file that contains an alternative index.

A, N/K. See *alphameric/Katakana.*

application. (1) A particular business task, such as inventory control or accounts receivable. (2) A group of related programs that apply to a particular business area, such as the Inventory Control or the Accounts Receivable application.

application program. A program used to perform an application or part of an application.

arithmetic operator. A symbol used to represent a mathematical operation, such as + or -, used to indicate addition or subtraction.

array. A named set of data items, all of which are the same type, arranged in a pattern (for example, columns and rows).

array element. A single data item in an array.

assignment. The process of giving values to variables.

attribute. A characteristic. For example, the attribute for a displayed field could be blinking.

authorize. To grant a user the right to communicate with or make use of a computer system or display station.

back up. To copy information, usually onto diskette, for safekeeping.

base number. The part of a self-check field from which the check digit is calculated.

binary. (1) Pertaining to a system of numbers to the base two; the binary digits are 0 and 1. (2) Involving a choice of two conditions, such as on-off or yes-no.

bit. Either of the binary digits 0 or 1. See also *byte.*

block. (1) A group of records that is recorded or processed as a unit. (2) Ten sectors (2560 bytes) of disk storage.

branching. Performing a statement other than the next one in sequence.

buffer. (1) A temporary storage unit, especially one that accepts information at one rate and delivers it at another rate. (2) An area of storage, temporarily reserved for performing input or output, into which data is read or from which data is written.

byte. The amount of storage required to represent one character; a byte is 8 bits.

call. To activate a program or procedure at its entry point.

CG indicator. See *current-group indicator.*

chain. A group of logically linked records.

chained fields. Fields that combine to form a single field that WSU uses (as the record key) to read or write a record in a master file.

character. A letter, digit, or other symbol.

character key. A keyboard key that allows the user to enter the character shown on the key. Compare with *command keys* and *function key.*

check. (1) An error condition. (2) To look for a condition.

check digit. The rightmost digit of a self-check field used to check the accuracy of the field.

code. (1) Instructions for the computer. (2) To write instructions for the computer. Same as *program.* (3) A representation of a condition, such as an error code.

collating sequence. The sequence in which characters are ordered within the computer for sorting, combining, or comparing.

column separator. A symbol on each side of a position of a field on a display. This symbol does not occupy a position on the display.

command. A request to perform an operation or a procedure.

command key indicator. Coded as KG through KL and KQ through KY. An indicator that is set on when an operator presses the corresponding command key.

command keys. The 12 keys on the top row of the display station keyboard that are used with the Cmd key (and optionally the Shift key) to request up to 24 different actions defined for program products and user programs. Compare with *character key* and *function key.*

comment. Words or statements in a program or procedure that serve as documentation rather than as instructions.

compile. To translate a program written in a high-level programming language into a machine language program.

concatenate. (1) To link together. (2) To join two character strings.

conditioning. The use of indicators to control when calculations or output operations are done.

conditioning indicator. An indicator used to indicate when calculations are done or which attributes apply to a format or format field.

configuration. The group of machines, devices, and programs that make up a data processing system. See also *system configuration.*

constant. A data item with a value that does not change. Contrast with *variable.*

continuation line. A line of a source statement into which characters are entered when the source statement cannot be contained on the previous line or lines.

creation date. The program date at the time a file is created. See also *program date*, *session date*, and *system date.*

current library. The first library searched for any required members. The current library can be specified during sign-on or while running programs and procedures.

current record. The record that is currently available to the program.

current-group indicator. Coded as CG. An indicator that signals whether the displayed record is from the same order set as the previously entered record.

cursor. A movable symbol (such as an underline) on a display, usually used to indicate to the operator where to type the next character.

data file utility (DFU). The part of the Utilities Program Product that is used to create, maintain, display, and print disk files.

DDM. See *Distributed Data Management.*

debug. To detect, locate, and remove mistakes from a program.

default prompt. A field name from a D-specification used to prompt for the field's contents.

default value. A value stored in the system that is used when no other value is specified.

delete. To remove. For example, to delete a file.

delete-capable file. A file from which records can be logically removed without compressing the file.

delete-mode indicator. Coded as DL. An indicator that signals that the transaction file record selected by the operator for review is to be logically deleted when the processing cycle ends.

detail record. A record that contains the daily activities or transactions of a business. For example, the items on a customer order are typically stored in detail records. Contrast with *header record.*

DFU. See *data file utility.*

diagnosed-source file. A library member containing source statements and associated error messages.

diagnostic. Pertaining to the detection and isolation of an error.

direct file. A disk file in which records are referenced by the relative record number. Contrast with *indexed file* and *sequential file.*

disk. A storage device made of one or more flat, circular plates with magnetic surfaces on which information can be stored.

diskette. A thin, flexible magnetic plate that is permanently sealed in a protective cover. It can be used to store information copied from the disk.

display. (1) A visual presentation of information on a display screen. (2) To show information on the display screen.

display format. Data that defines (or describes) a display.

display layout sheet. A form used to plan the location of data on the display.

display station. A device that includes a keyboard from which an operator can send information to the system and a display screen on which an operator can see the information sent or receive information from the system.

display station field. In WSU, a field that each display station uses and modifies independently.

display station indicator. In WSU, an indicator that each display station uses and modifies independently.

distributed data management (DDM). A part of the System Support Program Product that lets WSU use master and transaction files on a remote system.

DL indicator. See *delete-mode indicator.*

edit. (1) To modify the form or format of data; for example, to insert or remove characters such as for dates or decimal points. (2) To check the accuracy of information that has been entered, and to indicate if an error is found.

edit code. A number or letter indicating that editing should be done according to a defined pattern.

EJ indicator. See *end-of-job indicator.*

element. The smallest unit of data in a table or array.

embedded blanks. Blanks that are surrounded by any other characters.

end-of-job indicator. Coded as EJ. The indicator that signals that the last operator has specified the end of the work session on the WSU display. Compare with *end-of-work-session indicator.*

end-of-job processing level. The processing level that occurs once per job after the end of work session for the last display station or when the program sets the EJ indicator on. Compare with *end-of-work-session processing level.*

end-of-sequence-set indicator. Coded as ES. The indicator that signals that the last display in the primary sequence has been processed.

end-of-sequence-set processing level. The processing level that occurs after the last display in the primary sequence has been processed.

end-of-work-session indicator. The indicator that signals that an operator has specified end-of-work-session on the WSU display. Compare with *end-of-job indicator.*

end-of-work-session processing level. The processing level that occurs each time an operator specifies the end of the work session on the WSU display or when the program sets the EW or EJ indicator on. Compare with *end-of-job processing level.*

enter. To type in information on a keyboard and press the Enter key in order to send the information to the computer.

enter mode. The mode during which operators can add records to their transaction file.

ES indicator. See *end-of-sequence-set indicator.*

EW indicator. See *end-of-work-session indicator.*

execution-time array. An array that is loaded after the program begins.

external indicators. Indicators that can be set by another program before a program is run or changed while a program is running. The external indicators are U1 through U8.

extract. To obtain. For example, to extract information from a file.

factor. A field name, constant, literal, subroutine name, label, display name, or file name used in an operation.

field. One or more characters of related information (such as a name or an amount).

field area. An area in main storage that contains all of the fields defined in a WSU program.

file. A set of related records treated as a unit.

file definition. RPG II file description and input specifications that describe the records and fields in the transaction file and any master files.

file name. The name used by a program to identify a file. See also *label.*

first-level message. A message that communicates up to 75 characters of information to the operator. See also *second-level message.*

format. (1) A defined arrangement of such things as characters, fields, and lines, usually used for displays, printouts, or files. (2) To arrange such things as characters, fields, and lines.

format ID. Two characters that identify a display. An operator can enter this ID on the WSU display to select the display.

format member. A load member that contains display formats generated from S and D specifications in a program.

function key. A keyboard key that requests an action but does not display or print a character. The cursor movement and Help keys are examples of function keys. Compare with *command keys* and *character key.*

half-adjust. A method of rounding off a number by adjusting the last significant digit.

header record. A record that contains information, such as customer name and customer address, that is common to following detail records. Contrast with *detail record.*

hex. See *hexadecimal.*

hexadecimal. Pertaining to a system of numbers to the base sixteen; hexadecimal digits range from 0 (zero) through 9 (nine) and A (ten) through F (fifteen).

ID. Identification.

ideographic character (IGC). A two-position symbol, pictogram, or graphic character used to represent ideas.

ideographic support function. A version of SSP that manages alphameric-Katakana and ideographic-capable display stations, printers, and programs.

IFILE. See *immediate access file.*

IGC. See *ideographic character.*

IJ indicator. See *job-initiation indicator.*

immediate access file (IFILE). An index file to which records can be added by one program and then located and used immediately by another program without a key sort.

IN indicator. See *insert-mode indicator.*

indexed file. A file in which the key and the position of each record is recorded in a separate portion of the file called an index. Contrast with *direct file* and *sequential file.*

indicator. An internal switch that communicates a condition between parts of a program or procedure.

informational message. A message that provides information to the operator, but does not require a response.

initialize. To prepare for use. For example, to initialize a diskette.

initiate-transaction-sequence indicator. Coded as IS. An indicator that signals the start of a new function request.

input. Data to be processed.

input-to-process indicator. Coded as IP. In WSU, an indicator that signals whenever input data has been received from the previous display.

inquiry. (1) A request for information in storage. (2) A request that puts a display station into inquiry mode.

inquiry mode. A mode during which the job currently running from a display station is interrupted so that other work can be done. The operator puts the display station in inquiry mode by pressing the Attn key.

inquiry program. (1) A program that allows an operator to get information from a disk file. (2) A program that runs while the system is in inquiry mode.

insert mode. The mode during which operators can insert records in the transaction file.

insert-mode indicator. Coded as IN. The indicator that signals that the operator is running WSU in insert mode.

instruction. A statement that specifies an operation to be performed by the computer and the locations in storage of all data involved in that operation.

interactive. Pertains to activity involving requests and replies as, for example, between an operator and a program or between two programs.

IP indicator. See *input-to-process indicator.*

IS indicator. See *initiate-transaction-sequence indicator.*

IW indicator. See *work-session-initiation indicator.*

JA-JN and JP-JY indicators. See *job indicator.*

job. (1) A unit of work to be done by a system. (2) One or more related procedures or programs grouped into a procedure.

job control record. In WSU, the first record in the transaction file.

job indicator. Coded as JA through JN and JP through JY. An indicator that retains its setting throughout a job and is used by each display station running the job. Contrast with *mode indicator* and *session indicator.*

job queue. A list, on disk, of jobs waiting to be processed by the system.

job-initiation indicator. Coded as IJ. The indicator that signals that the first operator has signed on.

job-initiation processing level. The processing level that occurs only once per job when the first operator signs on.

job-level field. A field that remains in the field area in main storage. This field is available to any active display station that is using the WSU program. Contrast with *mode-level field* and *session-level field.*

K-byte. 1024 bytes.

Katakana. A native Japanese character set that is used to write foreign words phonetically in Japanese.

key. One or more characters used to identify the record and establish the record's order within an indexed file.

keyword. A symbol that identifies a parameter.

KG-KL and KQ-KY indicators. See *command key indicators.*

label. (1) The name in the disk or diskette volume table of contents that identifies a file. See also *file name.* (2) The name that identifies a statement.

LDA. See *local data area.*

left-adjust. To place or move an entry in a field so that the leftmost character of the field is in the leftmost position. Contrast with *right-adjust.*

library. (1) A named area on disk that can contain programs and related information (not files). A library consists of different sections, called library members. (2) The set of publications for a system.

library member. A named collection of records or statements in a library. The types of library members are *load member, procedure member, source member,* and *subroutine member.*

literal. A symbol or a quantity in a source program that is itself data, rather than a reference to data.

load member. A library member that contains information in a form that the system can use directly, such as a display format. Contrast with *source member.*

local data area (LDA). A 512-byte area on disk that can be used to pass information between jobs and job steps during a session. A separate local data area exists for each command display station.

local system. The system to which the user is currently signed on.

loop. A sequence of instructions that is performed repeatedly until an ending condition is reached.

magazine. A container that holds up to 10 diskettes.

main storage. The part of the processing unit where programs are run.

mandatory entry field. A field in which an operator must enter at least one character.

mandatory fill field. A field for which an operator must enter nothing or must fill in completely.

mask. A pattern of characters that controls the keeping, deleting, or testing of portions of another pattern of characters.

master file. A collection of permanent information, such as a customer address file.

member. See *library member.*

menu. A displayed list of items from which an operator can make a selection.

message. Information sent to an operator or programmer from a program. A message can be either displayed or printed.

message identification code (MIC). A four-digit number that identifies a record in a message member. This number can be part of the message identification.

MIC. See *message identification code.*

mode indicator. An indicator that changes when the operating mode changes. Contrast with *job indicator* and *session indicator.*

mode-level field. In a WSU program, a field that has its value saved and is then cleared when the operating mode changes. Contrast with *job-level field* and *session-level field.*

module. One part of a program, which usually performs a specific task (such as disk input/output).

modulus 10/modulus 11 checking. Formulas used to calculate the check digit for a self-check field.

MRT procedure. See *multiple requester terminal procedure.*

MRT program. See *multiple requester terminal program.*

multiple requester terminal (MRT) procedure. A procedure that calls a multiple requester terminal program.

multiple requester terminal (MRT) program. A program that can process requests from more than one display station or SSP-ICF session at the same time using a single copy of the program. Contrast with *single requester terminal (SRT) program.*

NEP. See *never-ending program.*

network resource directory (NRD). The part of Distributed Data Management for locating which remote system contains the file being accessed.

never-ending program (NEP). A long-running program that does not share system resources, except for shared files and the spool file.

non-contiguous key. A key that is made up of 2 or 3 non-consecutive record fields.

non-contiguous keyed file. An alternative indexed file with a non-contiguous key.

nondisplay. A field attribute that prevents the displaying of data.

nonsequenced display. A display that is not part of a sequence. Contrast with *primary display sequence, secondary display,* and *sequenced display.*

not-found indicator. An indicator that is set on when the specified record cannot be found.

NRD. See *network resource directory.*

null character. The character hex 00, used to represent the absence of a printed or displayed character.

numeric. Pertaining to any of the digits 0 through 9.

object program. A set of instructions in machine-runnable form. The object program is produced by a compiler from a source program.

OCL. See *operation control language.*

operand. A quantity of data that is operated on, or the address in a computer instruction of data to be operated on.

operation. A defined action, such as adding or comparing, performed on one or more data items.

operation code. A code used to represent the operations of a computer.

operation control language (OCL). A language used to identify a job and its processing requirements to the System Support Program Product.

operator. A person who operates a device.

output. The result of processing data.

overlay. (1) To write over (and therefore destroy) an existing file. (2) A program segment that is loaded into main storage and replaces all or part of a previously loaded program segment.

override. (1) A parameter or value that replaces a previous parameter or value. (2) To replace a parameter or value.

packed decimal format. A format in which each byte (except the rightmost byte) within a field represents two numeric digits. The rightmost byte contains one digit and the sign. For example, the decimal value +123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format.*

packed key. An index key in packed decimal format.

pad. To fill unused positions in a field with dummy data, usually zeros or blanks.

parameter. A value supplied to a procedure or program that either is used as input or controls the actions of the procedure or program.

PG indicator. See *program-mode indicator.*

position. The location of a character in a series, as in a record, a displayed message, or a computer printout.

preprocessed display. A display on which the processing is done before the display is shown.

preprocessing. Processing for a display that occurs before the display is shown.

primary display sequence. The first set of displays coded in a WSU source program.

printout. Information from the computer that is produced by a printer.

priority. The relative ranking of items. For example, a job with high priority in the job queue will be run before one with medium or low priority.

procedure. A set of related operation control language statements (and, possibly, utility control statements and procedure control expressions) that cause a specific program or set of programs to be performed.

procedure command. A command that runs a procedure.

procedure member. A library member that contains the statements (such as operation control language statements) necessary to perform a program or set of programs.

processing level. One of the stages in the WSU program cycle.

processing level display. A display that appears when the associated processing level occurs.

processing level indicator. The indicator (IJ, IW, ES, EW, or EJ) that WSU turns on automatically when the processing level begins, or that you can set on to cause any processing for that level to occur (ES, EW, EJ only).

program. (1) A sequence of instructions for a computer. See *source program* and *load module.* (2) To write a sequence of instructions for a computer. Same as *code.*

program date. The date associated with a program (job step). See also *creation date,session date,* and *system date.*

program generation. The compilation of a WSU program.

program-mode indicator. Coded as PG. An indicator that temporarily prevents operator-initiated changes in transaction file contents or display sequence.

program product. A licensed program for which a fee is charged.

prompt. A displayed request for information or operator action.

protected field. A displayed field in which operators cannot enter data.

quotient. The number or quantity that is the result of a division operation.

RC indicator. See *recovery-of-work-session indicator.*

record. A collection of fields that is treated as a unit.

record identification code. Characters placed in a record to identify that record type.

record identifying indicator. An indicator that identifies the record just read.

record type. The classification of records in a file.

recovery-of-work-session indicator. Coded as RC. The indicator that signals when an operator restarts a work session after abnormal termination.

relative record number. A number that specifies the location of a record in relation to the beginning of the file.

remote file. A file that resides on a remote system.

remote system. A system that is connected to the local system via a communication facility.

repeated-display indicator. Coded as RP. The indicator that signals when a display reappears.

reserved fields. Special fields provided and maintained by WSU that contain such current information as relative record numbers, date and time, and error codes.

resulting indicator. An indicator that is set depending on the result of an operation.

resume-work-session indicator. Coded as RS. The indicator that signals when an operator resumes running WSU after all other operators successfully ended their work sessions.

review mode. The mode of WSU in which operators can selectively display transaction file records or can move from one transaction file record to another.

review-mode indicator. Coded as RV. The indicator that signals that the operator is running WSU in review mode or in insert mode.

review-record indicator. Coded as RR. The indicator that signals review or delete.

right-adjust. To place or move an entry in a field so that the rightmost character of the field is in the rightmost position. Contrast with *left-adjust.*

routine. A set of statements in a program that causes the system to perform an operation or a series of related operations.

rover-update indicator. Coded as RU. An indicator that determines whether the transaction file can be updated.

RP indicator. See *repeated-display indicator.*

RPG. A programming language specifically designed for writing application programs that meet common business data processing requirements.

RR indicator. See *review-record indicator.*

RS indicator. See *resume-work-session indicator.*

RU indicator. See *rover-update indicator.*

RV indicator. See *review-mode indicator.*

SA-SN and SP-SY. See *session indicator.*

screen design aid (SDA). The part of the Utilities Program Product that helps the user design, create, and maintain displays and menus. Additionally, SDA can generate specifications for RPG and WSU work station programs.

SDA. See *screen design aid.*

secondary display sequence. The set of displays that follows the primary display sequence in a WSU source program.

sector. (1) An area on a disk track or a diskette track reserved to record information. (2) The smallest amount of information that can be written to or read from a disk or diskette during a single read or write operation.

self-check field. A field, such as an account number, consisting of a base number and a check digit.

sequence checking. Checking done by WSU (in enter mode only) to ensure that operators use displays in the intended (coded) order. For example, WSU checks to ensure that operators do not bypass required displays.

sequence error. In WSU, an error caused by trying to bypass required displays or end-of-sequence-set processing.

sequence number. A five-digit entry on specifications that indicates the order of the specifications.

sequence set. In WSU, one or more related displays that appear in the order in which they occur in the source program.

sequenced display. A display within a sequence. See *nonsequenced display.*

sequential file. A file in which records occur in the order in which they were entered. Contrast with *direct file* and *indexed file.*

serialization. The consecutive numbering of statements.

session. (1) The logical connection by which a System/36 program or device can communicate with a program or device at a remote location. (2) The length of time that starts when an operator signs on the system and ends when the operator signs off the system.

session control record. The first record in each display station's chain of records in the transaction file.

session date. The date associated with a session. See also *creation date, program date,* and *system date.*

session indicator. An indicator that is set for each session and retains its setting when the operating mode changes. Contrast with *job indicator* and *mode indicator.*

session-level field. A field that retains its value when the operating mode changes. Contrast with *job-level field* and *mode-level field.*

SEU. See *source entry utility.*

severity code. A code that indicates how serious a compiling or an operating error is.

shift-in (SI) character. A one-position character (0F) that signifies the end of an ideographic character string.

shift-out (SO) character. A one-position character (0E) that signifies the start of an ideographic character string.

SI. See *shift-in*(0F)

SO. See *shift-out* (0E)

sign off. To end a session at a display station.

sign on. (Verb) To begin a session at a display station.

single requester terminal (SRT) program. A program that can process requests from only one display station or SSP-ICF session from each copy of the program. Contrast with *multiple requester terminal program.*

sort utility. The part of the System Support Program Product used to arrange records (or their relative record numbers) in a sequence determined by data contained in one or more fields within the records.

source entry utility (SEU). The part of the Utilities Program Product used by the operator to enter and update source and procedure members.

source member. A library member that contains information in the form in which it was entered, such as RPG specifications. Contrast with *load member.*

source program. A set of instructions that are written in a programming language and that must be translated to machine language before the program can be run.

source statement. A statement written in a programming language.

special character. (1) A character other than an alphabetic or numeric character. For example; *, +, and % are special characters. (2) A COBOL character that is neither numeric nor alphabetic. Special characters in COBOL include the space (), and the period (.), as well as the following: + - * / = $, ") (; < >.

specification display. In SEU, the display that guides an operator through the entry of a particular type of statement.

specification sheets. Forms on which a program is coded and described.

specification statement. One of the set of statements that provide the compiler with information about the data used in the source program. In addition, the statement supplies information required to reserve storage for this data.

SRT program. See *single requester terminal program.*

SSP. See *System Support Program Product.*

statement. An instruction in a program or procedure.

statement number. The number assigned to each specification. Comments, continued lines, F specifications, and unrecognized or out-of-sequence specifications are not assigned statement numbers.

status line. Line 1 of the display which contains general information about a job.

subroutine. A group of instructions that can be called by another program or subroutine.

subroutine member. A library member that contains information that must be combined with one or more members before being run by the system.

subscript. An integer or variable whose value refers to a particular element in a table or an array.

syntax. The rules for the construction of a statement.

system. The computer and its associated devices and programs.

system configuration. A process that specifies the machines, devices, and programs that form a particular data processing system.

system console. A display station from which an operator can keep track of and control system operation.

system date. The date assigned by the system operator during the initial program load procedure. See also *creation date, program date,* and *session date.*

system library. The library, provided with the system, that contains the System Support Program Product and is named #LIBRARY.

System Support Program Product (SSP). A group of licensed programs that manage the running of other programs and the operation of associated devices, such as the display station and printer. The SSP also contains utility programs that perform common tasks, such as copying information from diskette to disk.

table. A collection of data in which each item is uniquely identified by a label, by its position relative to the other items, or by some other means.

task. A unit of work (such as a user program) for the main storage processor.

terminal error. In WSU, an error that requires correction.

track. A circular path on the surface of a disk or diskette on which information is magnetically recorded and from which recorded information is read.

trailer. Control information that WSU adds to the end of each record in a transaction file.

transaction. An item of business. The handling of customer orders and customer billing are examples of transactions.

transaction file. (1) A file containing data, such as customer orders, that is usually used only with a master file. (2) In WSU, a direct file containing control records and data records for each work session.

truncate. To shorten a field or statement to a specified length.

unique. The only one.

unprotected field. A displayed field for which operators can enter, modify, or delete data.

UPSI switch. See *user program status indicator switch.*

user program status indicator (UPSI) switch. One of a set of eight switches that can be set by and passed between application programs and procedures.

Utilities Program Product. A program product that contains the data file utility (DFU), the source entry utility (SEU), the work station utility (WSU), and the screen design aid (SDA).

utility control statement. A statement that gives a utility program information about the way the program is to perform or the output it is to produce.

utility program. A System Support Program Product program that allows you to perform a common task, such as copying information from diskette to disk.

U1-U8 indicators. See *external indicators.*

variable. A name used to represent a data item whose value can change while the program is running. Contrast with *constant.*

work file. A file that is used for temporary storage of data being processed.

work session. The time during which an operator is using a WSU program.

work session control record. The first record in each work session's chain of records in the transaction file.

work station. A device that lets people transmit information to or receive information from a computer; for example, a display station or printer.

work station utility (WSU). The part of the Utilities Program Product that helps you to write programs for data entry, editing, and inquiry.

work-session-initiation processing level. The processing level that occurs once when each operator starts a WSU program. For the first display station operator, the work session begins when job has been started.

work-session-initiation indicator. Coded as IW. The indicator that is set when an operator starts a WSU program.

WSU. See *work station utility.*

WSU display. A display selected by the WSU display command key that allows operators to select a display, review a record, or end a work session.

WSU-generated procedure. The procedure that WSU creates to load and run a WSU program for the first operator who calls the procedure.

zero suppression. The substitution of blanks for leading zeros in a number. For example, 00057 becomes 57 when using zero suppression.

zoned decimal format. A format for representing numbers in which the digit is contained in bits 4 through 7 and the sign is contained in bits 0 through 3 of the rightmost byte; bits 0 through 3 of all other bytes contain 1s (hex F). For example, in zoned decimal format, the decimal value of +123 is represented as 1111 0001 1111 0010 1111 0011. Contrast with *packed decimal format.*

Index

Special Characters

- *BLANK
  - use in MOVE operation 3-6
  - where coded 3-6
- *ERROR
  - error codes 3-4
  - error indicator on m-specification 15-8
  - error indicator on t-specification 15-4
  - how to code 3-4
- *RLNO
  - how to code 3-3
  - use in transaction file 3-3
- *RLRN
  - how to code 3-5
  - use as GET field 3-5
- *RLRR
  - how to code 3-5
  - initial value 3-5
- *RLRU
  - how to code 3-5
  - input operations 3-5
- *SLNO
- *USID
  - how to code 3-6
  - use in operations 3-6
- *WSID
  - how to code 3-6
  - use on C-specifications 3-6

A

- accept-command-key indicator
  - set off 3-15
  - set on 3-15
  - when used 17-3
  - where coded 3-15
- accept-sequence-error indicator
  - on C-specification 17-3
  - set off 3-15
  - set on 3-15
  - when used 17-3, 17-4
  - where coded 3-15
- ADD
  - explanation 7-11
  - uses 18-1
- adjust/fill, restrictions 16-21
- alphabetic input field 16-18
- alphameric data to numeric data 18-16, 18-18
- alphameric fields, banking 18-16
- alternative indexed files 4-1
- AN/OR, when used 17-2
- AND 4-14
- AND line, maximum number 4-14
- AND/OR relationships
  - AND 4-14
  - OR 4-15
- AND/OR restriction 3-29
- arithmetic operations 17-11
  - ADD 18-1
  - DIV (divide) 18-4
  - half-adjust 17-11
  - MULT (multiply) 18-22
  - SUB (subtract) 18-29
  - Z-ADD (zero and add) 18-30
- array columns 16-15
- array element
  - maximum number 14-1
- array entry length
  - binary format 14-2
  - numeric 14-2
  - packed decimal 14-2
- array name, restrictions
- array operations
  - examples 5-7
  - LOKUP (look for next element of array) 18-14
  - modifying contents of 5-5
  - SORTA (sort array) 18-28
- array rows 16-15
- array types
  - alphameric array
  - numeric array
    - binary format 5-2
    - packed decimal format 5-2
- arrays
  - accumulated totals 5-8
  - adding all elements 5-14
  - array entries on D-specification 6-15
  - array-to-array processing 5-7
  - C-(processing) specifications 5-21
  - characteristics 5-1
  - D-specifications 5-30
  - decimal positions on 5-34
  - defining 5-5
  - different length arrays, processing 5-10
  - displaying 6-15
    - entire array 5-30
    - individual elements 5-31
  - E-(array) specifications 5-5
  - elements
    - consecutive positions 5-33
    - displaying 5-30
    - I-specification entries from scattered positions 5-36
    - maximum number of 5-6
    - reading from consecutive positions 5-34
    - scattered positions 5-33
  - field scanning 5-19
  - index
    - constant 5-16
    - to reference 5-6
    - variable 5-18
  - index fields 5-25, 5-26
  - indexing 5-15
  - job level 5-5
  - length
    - alphameric 5-6
    - numeric 5-6
  - loading 5-32
  - maximum number 1-7
  - mode-level 5-5
  - naming of 5-6
  - numeric 5-1
  - operations on
    - arithmetic 5-7
    - COMP 5-15, 5-23
    - LOKUP 5-6, 5-7, 5-23, 5-28
    - MOVE 5-7
    - MOVEL 5-7
    - MVR 5-15
    - SORTA 5-6, 5-15, 5-22
    - XFOOT 5-7, 5-14, 5-15, 5-16
    - Z-ADD 5-36
  - Processing
    - array-to-array 5-7
    - different length 5-10
    - single-value arrays 5-11
  - reading data into 5-32
  - referencing
    - elements 5-25
    - entire array 5-7
    - individual elements 5-15
    - part of a field 5-19
  - result field name 5-34
  - resulting indicators 5-7, 5-14, 5-25
  - rolling totals 5-9
  - scanning 5-19
  - searching

- for a particular element 5-21
- more than one element 5-27
- results, displaying 5-29
- starting at a particular element 5-23
- successful search 5-25
- sequence checking 5-6
- session-level 5-5
- single-value arrays, processing 5-11
- spacing of 5-6
- using arrays
  - totals 5-2
  - variable data 5-2
- XFOOT (summing values of 18-30
- ascending sequence, arrays
- assigning priority B-1
- assumed starting line number 16-3
- attention key, to end a work station 10-14
- attribute
  - invalid combinations 16-27
- authorized display stations 10-2

B

- batch-edit 2-14
- BEGSR
  - explanation 7-33
  - uses 18-1
- binary format 4-19
- blanks, changed to nulls 16-9
- blink field, restrictions 16-25
- blocks needed for WSU program 8-12
- branching, operations
  - GOTO, explanation 7-19
  - GOTO, requirements 18-12
  - TAG, explanation 7-19
  - TAG, requirements 18-29
- bypass display 10-4
- Bypass Display command key 16-10
- bypass mandatory-entry fields 16-4

C

- C-specification entries
  - comments 7-6, 17-14
  - conditioning indicators 17-3
  - Decimal Positions 7-5, 17-10
  - Factor 1 7-4, 17-5
  - Factor 2 7-4, 17-5
  - Half Adjust 7-5
  - half-adjust 17-11
  - length 17-9
  - literals 17-6
  - name 17-9
  - operation 17-6
  - Processing Function 7-3, 17-2
  - RANGE operation 17-6
  - Result Field 7-5, 17-9
  - Result Field Length 7-5
  - Resulting Indicators 7-6
    - plus or high 17-12
    - zero or equal 1=2 17-12
  - table or message text 17-6
- chain of records 10-2
- chain-end indicator field
- characters, using ideographic 20-1
- check digit
- CHN information 9-10
- coding forms
  - RPG F-(File description) specification 12-1
  - RPG I-(Input) Specifications 12-4
  - S/36 S- and D- (Display Format) Specifications 16-1
  - S/36 WSU c-(processing) 17-1
  - S/36 WSU job, array, and file specification 13-1
- coding, files 4-25
- column separators
  - coding 16-28
  - restrictions 16-28
- command key indicators
  - key/indicator list 3-22
  - set off 3-22
  - set on 3-22
  - when used 17-3
  - where coded 3-22
- command keys
  - disabling 16-13
  - enabling 16-7, 16-13
- command statement
  - WSUTXCR 19-7
  - WSUTXEX 19-1
  - WSUTXRV 19-12
- comments, on WSU program 17-14
- COMP
  - comparing and testing operations 7-20
  - comparing factor 1 to a field or literal 7-20
  - comparing factor 1 to a table 18-2
- comparing and testing operations
  - COMP, explanation 7-20
  - COMP, requirements 18-2
  - RANGE, explanation 7-20
  - RANGE, requirements 18-8
- comparing factor 1 to field or literal 7-20
- comparing factor 1 to table 7-22
- COMPILE menu 8-9
- concatenation 20-2
- conditioning indicators
  - AC 7-3, 17-3
  - AE 7-3, 17-3
  - Blank 7-3, 17-3
  - CG 7-3, 17-3
  - DL 7-3, 17-3
  - EJ 7-3, 17-3
  - ES 7-3, 17-3
  - EW 7-3, 17-3
  - IJ 7-3, 17-3
  - IMSG operation 18-13
  - IN 7-3, 17-3
  - IP 7-3, 17-3
  - IS 7-3, 17-3
  - IW 7-3, 17-3
  - JA-JN, JP-JY 7-3, 17-3
  - KG-KL, KQ-KY 7-3
  - KG-KN, KP-KY 17-3
  - linking 17-2
  - MSG operation 18-20
  - PG 7-4, 17-3
  - RC 7-4, 17-3
  - RP 7-4, 17-3
  - RR 7-4, 17-3
  - RS 7-4, 17-4
  - RU 7-4, 17-4
  - RV 7-4, 17-4
  - SA-SN, SP-SY 7-4
  - using more than three 17-4
  - U1-U8 7-4, 17-4
  - 01-89 7-4
- constant
  - constant type and constant data 6-14
  - data 6-10
  - fields 6-9
  - types 16-30
- constant value
  - constant data 6-10
  - for input fields 6-10
  - initial 16-30
  - on D-specification 16-30
- continuation of data coding 16-32
- controlled field exit 16-22
- current display station 2-23
- current library 13-2
- current-group indicator
  - on C-specifications 17-3
  - set off 3-16
  - set on 3-16
  - to get WSU aid display 10-8
  - use of 10-4

  - when used 17-3
  - where coded 3-16
- cursor
  - blinking 16-8
  - movement 10-10
  - positioning 16-21

D

- D-(field description) specification 6-10
- D-CONT 8-8
- D-specification entries
  - adjust/fill 6-12, 16-21
  - auto record advance 6-11, 16-23
  - blink field 6-14
  - column separator 6-14
  - column separators 16-28
  - constant data 6-14, 16-30
  - constant type 6-14, 16-30
  - continuation 6-14, 16-32
  - controlled field exit 6-11
  - d-specification entries 16-15, 16-16
  - data type 6-12, 16-18
  - edit code 6-12, 16-17
  - elements per row or column 6-15
  - fill direction 6-15
  - high intensity 6-14, 16-24
  - horizontal positions space factor 6-15
  - input allowed 6-10, 16-17
  - line skip factor 6-15
  - lowercase 6-12, 16-29
  - mandatory entry 6-12
  - mandatory fill 6-12, 16-20
  - nondisplay 6-14, 16-25
  - output data 6-10, 16-17
  - position cursor 6-11, 16-22
  - protect field 6-12, 16-23
  - reverse image 6-14, 16-27
  - self check 6-15
  - self-check 16-21, 16-22
  - starting location 6-10
  - underline 6-14, 16-27
  - WSU arrays 16-15
  - WSU field name 6-10
- D-specification, required entries 6-10
- data
  - conversion 4-18, 4-20
  - directing 7-1
  - entering 10-10
  - entry with edit 1-6
  - entry with update 1-6
  - handling 7-1
  - how represented in computer 4-16
  - maximum length for compared alphameric 7-20
  - maximum length for compared numeric 7-20
  - underlined 16-27
- data formats
  - binary format 4-19
  - examples 4-17
  - on I-specification 4-16
  - packed decimal format 4-18
  - zoned decimal format 4-16
- data management, extended B-3
- data type
  - ideographic 16-18
- data verification, numeric data 4-16
- date fields
  - as GET fields 3-2
  - as output fields 3-2
  - list 3-2
  - table of values 3-2
  - uses 3-2
  - WSU reserved fields 3-1
- DDM (Distributed Data Management) 1-7, 4-1, 4-2, B-3
- DEBUG
  - on C-specification 7-24
  - requirements 18-2
  - uses 18-2
- debugging
  - a WSU program 9-1
  - display 18-3
  - operations 18-3
- DEC information 9-10
- decimal alignment 7-11
- decimal positions
- delete indicator
  - when used 17-3
- delete mode 2-4, 2-21
  - cancelling delete request 2-21
  - delete mode 10-4
  - mode of operation 2-4
  - transaction file chain 2-21
  - using 2-21
  - WSU's role
    - determine most recently reviewed record 2-22
    - search the S-specification 2-22
    - send a message 2-22
    - set on and off delete mode indicator 2-22
    - show the most recently reviewed record 2-22
    - update the trailers 2-22
- delete-capable file considerations B-3
- delete-mode indicator
  - set off 3-16
  - set on 3-16
  - where coded 3-16
- diagnostic
  - checklist 9-2
  - data field names used 9-10
  - displayed text for programmed 9-11
  - extended diagnostics, cross-reference 9-8
  - field name and label usages 9-9
  - generation 9-2
  - in running 9-2
  - indicator name usage, alphabetic 9-8
  - main storage requirements 9-12
  - text 9-12
  - tools 9-3
  - user message member MICs used 9-9
- diagnostic information
- DIM information 9-10
- direct files
  - master files 4-1
  - transaction files 4-2
- disk space
  - allocation 15-4
  - saving 4-18
- disk storage requirements 9-12
- DISP information 9-10
- display characteristics
  - each of the fields 6-1
  - entire display 6-1
- display format information
  - execution time input buffer 9-13
  - execution time output buffer 9-13
  - format load member name 9-13
  - input library name 9-13
  - main storage required 9-13
  - name of source member 9-13
  - on generated output 9-13
  - output library name 9-13
  - S- and D- specifications 9-13
- display formats
  - load member 8-17
  - where stored 13-1
- display processing
  - allowed 16-5
  - not allowed 16-5
  - Reset Keyboard 16-5
- display selecting
  - controlling 2-12
  - from WSU display 6-3, 10-6
- display sequences 2-7
  - Bypass Display command key 2-7
  - consists of 6-5
  - defining 2-8
    - figure 2-8
  - end of 6-5

  frequency of use 6-7
  in review mode 2-12
  modifying 2-7
  nonsequenced 2-7
  primary 2-7
  secondary 2-7
  start of 6-5
display sequences in review mode 2-12
display station
  changing maximum number of 13-3
  current 2-23
  maximum number 8-18, 8-19
display station fields
  *RLNO 3-3
  *SLNO 3-3
  mode-level fields 2-23
  session-level fields 2-23
  when used 17-4
display station identifier, symbolic 3-6
display station keyboard, illustration 10-3
display type Z 8-19
displays
  attributes 6-4
  automatic advance 16-22
  bottom line use 16-3
  Bypass Display command key 2-7
  coding 6-1
  guidelines 6-1
  how field will look 6-14
  input fields 6-9
  maximum number per program 6-9
  number of 2-7
  output fields 6-9
  output/input fields 6-9
  primary sequence 16-9
  returning to 10-8
  review mode displays 2-11
    combined enter/review displays 2-11
    separate review mode displays 2-11
  secondary sequence 16-9
  select alternate 10-12
  sequenced 2-7
Distributed Data Management (DDM) 1-7, 4-1, 4-2, B-3
DIV
  explanation 7-11
  requirements 18-4
duplicate format names 16-1

E

E-specification
  array name 14-1
  decimal positions 14-1
  length of entry 14-1
  location in program 14-1
  number of entries in the array 14-1
  sequence A/D 14-1
  uses 5-8
edit code
  restrictions 16-17
end-of-job indicator
  on C-specification 17-3
  set off 3-17
  set on 3-17
  when used 17-2
  where coded 3-17
end-of-processing routine, common 7-9
end-of-sequence-set indicator
  set off 3-18
  set on 3-18
  when used 17-2, 17-3
  where coded 3-18, 17-3
end-of-work-session indicator
  on C-specifications 17-3
  set off 3-19
  set on 3-19
  when used 17-2, 17-3
  where coded 3-19
End, restrictions 16-9
ENDSR
  explanation 7-33
  in subroutines 7-7
  requirements 18-4
enter mode 2-4
  Bypass Display command key 2-7
  display sequences 2-7
  figure 2-6
  processing levels 2-5
  program cycle 2-4
  programming cycle 2-4
  return to 10-12
Enter Mode Sequence, restrictions 16-9
error codes 3-4
error conditions 9-15
errors
  finding and correcting 9-1
  program 9-1
execution time input buffer 9-13
execution time output buffer 9-13
EXSR
  in subroutines 7-7
  requirements 18-5
  uses 7-33
extend file considerations B-3
extended data management B-3
external indicators
  set off 3-27
  set on 3-27
  where coded 3-27
  01-89 indicators 3-28

F

F- and I-specifications, sample coding 4-21
F-specification, entries
  File name 12-2
  File Type 12-2
  Key Field Starting Location 12-3
  Length of Key Field 12-3
  Record Address Type 12-3
  Record Length 12-2
  required entries 4-11
    file name 4-11
    file type 4-11
    record length 4-11
  required for indexed master file 4-11
    key field length 4-11
    key field start location 4-11
    record address type 4-11
  Type of File Organization 12-3
F-specifications
  coding example 4-21
  in file definition 8-4
F/ICNT information 9-10
factor 1 and factor 2 entries, chart of
Factor 1 and Factor 2, comparing 18-2
Factor 1, comparing to field or literal 7-20
field exit keys
  enter a negative value 10-10
field lengths
  alphameric 12-8
  binary 12-8
  maximum 7-6
  packed numeric 12-8
  zoned decimal 12-8
field lines 4-13
field name and label usages 9-9
field names
  defined more than once 9-11
  undefined 9-11
  unreferenced 9-11
field+ key 16-18
fields
  blinking 16-25
  describe 15-8
  display station 2-23

  - how will look on display 6-14
  - in main storage 3-1
  - input 6-9, 6-11
  - main storage 3-1
  - mode level 3-1
  - nondisplay 16-25
  - numeric output 6-12
  - output 6-9, 6-11
  - output/input 6-9, 6-11
  - packed on I-specifications 7-6
  - protected 16-23
  - saving 2-16
  - session level 3-1
  - skipping over 10-10
  - unprotected 16-23
- file coding 4-25, 8-4
- file coding, included in 4-25
- file definition
  - coding forms 4-10
  - contents 4-10
  - entering 8-2
  - F-specification 4-11
  - I-specification 4-12
  - in a WSU program 1-8
  - storing 8-4
- file disposition 8-22
- file processing procedures 4-5
- file type
  - direct output 12-2
  - input 12-2
- format ID
  - restrictions 16-2
- Format ID, processing level indicators
  - EJ 16-2
  - EW 16-2
  - IJ 16-2
  - IW 16-2
- format load member 9-13
- function keys 10-5
  - enabling 16-13
  - page backward record 10-5
  - page forward record 10-5
  - to enable 16-6

G

- generation message 9-12
- generation output
  - created by 8-17
  - example of 9-4
  - explanation 8-17
- generation printout
  - created by 8-17
  - example of 9-4
  - explanation 8-17
- generation process 8-16
- GET
  - combining of 15-7
  - read a master file record 7-25
  - uses 18-6
- GETNH
  - explanation 7-25
  - requirements 18-7
  - transaction file 18-7
- GETNR
  - explanation 7-25
  - requirements 18-8
- GETPH
  - explanation 7-25
  - requirements 18-9
- GETPR
  - explanation 7-26
  - requirements 18-11
- GOTO
  - in subroutines 7-7
  - requirements 18-12
  - skip instructions 7-19

H

- header record identifying, use to 15-3
- heading information 9-4
- help displays
  - explanation B-5
  - requirements B-5
  - usage B-5
- history file logging 8-20
- horizontal position, restrictions 16-16
- how to code 3-3

I

- I-specification
  - coding example 4-21
  - in file definition 8-4
  - location 12-4
  - required entries 4-12
- I-specification entries
  - AND/OR 4-12, 12-4
  - C/Z/D 12-6
  - Character 12-6
  - decimal positions 4-13, 12-8
  - field location 4-13, 12-7
  - field name 4-13, 12-8
  - file name 4-12, 12-4
  - Not 12-5
  - P/B/L/R 4-13, 12-7
  - Position 12-5
  - record identification codes 4-12, 12-5
  - record identifying indicator 4-12, 12-5
- I-specification file name, entries 12-4
- ideographic
  - characters 20-1
  - concatenation 20-2
  - data type 16-18
  - restrictions 20-2
  - space requirements 20-1
- IFILE (immediate access file) 19-1, 19-7
- immediate access file (IFILE) 19-1, 19-7
- IMSG
  - explanation 7-26
  - requirements 18-13
- indexed files
  - master files 4-1
- indicator coding chart 3-30
- indicator name usage 9-9
- indicator operations
  - SETOF (set off) 18-27
  - SETON (set on) 18-28
- indicators
  - AC 3-15
  - AE 3-15
  - assign 17-4
  - CG 3-16
  - command key 3-7
  - DL 3-16
  - EJ 3-17
  - ES 3-18
  - EW 3-19
  - external 3-27
  - first called 3-14
  - IJ 3-14, 3-19
  - IN 3-20
  - initial settings 3-14
  - IP 3-20
  - IS 3-20
  - IW 3-14, 3-21
  - JA-JN, JP-JY 3-21
  - job 3-9
  - KG-KL, KQ-KY 3-22
  - mode 3-12
  - operating mode 3-7
  - PG 3-23

  - processing level 3-7
  - RC 3-23
  - RP 3-24
  - RR 3-25
  - RS 3-26
  - RU 3-26
  - RV 3-26
  - SA-SN, SP-SY 3-27
  - session 3-9
  - setting on and off 3-7
  - summary table 3-14
  - use of 17-3
  - U1-U8 3-14, 3-27
  - when used 17-3, 17-4
  - where to code chart 3-30
  - 01-89 3-28
- indicators, programmable
  - AC 3-8
  - AE 3-8
  - EJ 3-8
  - ES 3-8
  - EW 3-8
  - IS 3-8
  - JA-JN, JP-JY 3-8
  - KG-KL, KQ-KY 3-8
  - PG 3-8
  - RC 3-8
  - RP 3-8
  - RS 3-8
  - SA-SN, SP-SY 3-8
  - U1-U8 3-8
  - 01-89 3-8
- indicators, session-level
  - AE 3-10
  - ES 3-10
  - EW 3-10
  - indicators 3-27
    - set off 3-27
    - set on 3-27
    - where coded 3-27
  - IW 3-10
  - RC 3-10
  - RS 3-10
  - SA-SN, SP-SY 3-10
  - U1-U8 3-10
- initial value 3-3
- initialize values 7-9
- initiate-transaction indicator
  - on C-specification 17-3
  - set off 3-20
  - set on 3-20
  - when used 17-4
  - where coded 3-20
- input and output operations
  - GET 18-6
  - GETNH 18-7
  - GETNR 18-8
  - GETPH 18-10
  - MSG 18-20
  - PUT 18-24
  - PUTN 18-24
  - PUTS 18-25
  - TIME 18-30
- input fields
  - data fields 6-9
  - mandatory fill 16-20
  - maximum number 6-9
- input library name 9-13
- input-to-process indicator
  - on C-specification 17-3, 17-6
  - set off 3-20
  - set on 3-20
  - when used 17-3
  - where coded 3-20
- inquiry display 10-14
- inquiry program
  - with master files 1-4
- insert mode 2-4, 2-18
  - Bypass Display command key 2-19
  - display sequence 2-19
  - insert mode 10-4
  - mode of operation 2-4
  - transaction file 2-18
  - using 2-18
  - what WSU can do in 2-20
  - with review mode 2-18
  - WSU's role
    - place the inserted record 2-20
    - present the display 2-20
    - read displayed record 2-20
    - search the S-specifications 2-20
    - send message to operator 2-20
    - set on and off insert mode indicator 2-20
    - updates trailers 2-20
- Insert Mode Record Identifying Indicators, restrictions 16-11
- insert-capable 2-18
- insert-mode indicator
  - on C-specification 17-3
  - on S-specification 16-11
  - set off 3-20
  - set on 3-20
  - when used 17-3
  - where coded 3-20
- inserting a group of records 2-19
- instruction
  - skipping 18-12
- intensify data 16-24
- interactive communications feature B-3
- internal switch 3-7

J

- J-specification
  - coding 4-26
  - location 13-1
  - number per program 4-25
  - required entries 4-26
- J-specification, entries
  - date/edit 13-4
  - display format member name 4-26, 13-1
  - library name 13-2
  - maximum number of display stations 13-3
  - message load member name 13-2
  - program name 4-26, 13-1
  - region size 13-3
- JA-JN, JP-JY (Job) Indicators
  - set off 3-21
  - set on 3-21
  - where coded 3-21
- JA-JN, JP-JY indicators 3-9
- job
  - explanation 10-2
  - how WSU works 2-1
- job coding 4-26
- job control record
  - in transaction files 4-3
- job field, define 17-9
- job indicators
  - EJ 3-9
  - IJ 3-9
  - JA-JN, JP-JY 3-9, 3-21
    - set off 3-21
    - set on 3-21
    - where coded 3-21
  - list 3-9
- job-initiation-processing indicator
  - on C-specification 17-3
  - set off 3-19
  - set on 3-19
  - when used 17-3
  - where coded 3-19
- job-level fields
  - job-level fields 2-23, 9-10

K

key mask
  chart 16-12
  ID 16-12
  key mask 16-12
  list 16-12

L

LCL information 9-10
library name
  in file definition 8-2
  in source program 8-5
  on WSU-generated procedure 8-18
literals
  alphameric 17-6
    apostrophes 17-6
    requirements 17-6
  numeric 17-6
    apostrophes 17-7
    maximum length 17-7
    requirements 17-7
    rules 17-6
  use in operations 17-5
LNG information 9-10
load member
  containing display formats 8-18
  containing messages 8-18
  on J-specification 13-1
local data area (LDA)
  a file name 15-5
  describe 15-8
  in file definition type 15-8
  use in WSU program 2-24
logical delete considerations B-4
LOKUP
  high or low 14-2
  in array operations 7-13
  uses 18-14
lowercase
  D-specifications 16-29
  priority 16-4, 16-29
  restrictions 16-4
  S-specifications 16-4

M

M-specification
  coding 4-29
  error indicator 15-7
  field level 15-9
  file definition location 15-6
  file definition type 15-8
  file name 15-5
  get field names 15-7
  library name 15-6
  location 15-5
  not-found indicators 15-7
  number per program 4-25
  source member name 15-6
  use of 4-29
M-specification entries
  error indicator 4-30
  field level 4-30
  file definition type 4-30
  file name 4-29
  GET field names 4-29
  library name 4-29
  not-found indicator 4-30
  source member name 4-29
main storage
  additional room B-1
  amount 13-3
  changing amount of 13-3
  DEBUG operation 18-2
  requirements 8-21, 9-12
maintenance programs 1-4
mandatory entry, restrictions 16-20
master fields, describe 15-8
master file
  alternative index 4-1
  error codes 3-4
  file organization 12-3
  indexed 4-11, 12-3
  indexed on PUTN operation 18-24
  input file 12-2
  key field length 12-3
  key field starting location 12-3
  maximum entry 12-7
  maximum length 12-5
  maximum number 1-7, 4-1
  name 8-18, 15-5
  non-contiguous keys 4-1, 4-11, 12-3
  physical characteristics 4-1
  read record from 4-1
  record address type 12-3
  record length 12-2
  remote files 1-7, 4-1, B-3
  sharing 4-2
  update file 12-2
  updating a shared 4-2
  updating during review mode 2-17
  use of 1-4
  uses 4-1
  write records to 4-1
master file records, updating 2-17
member name
  in source program 8-5
member type 8-2
  in source program 8-5
message 2-16
  send a 2-16
  where stored 13-2
message displaying 16-3
message identification code 16-31
message load member
  J-specification 13-1, 13-2
  library name 8-18
  name and library 8-20
  where stored 13-2
message overlaying 18-20
message text
  continuation 17-11
  IMSG operation 17-6, 18-13
  MSG operation 17-6, 18-21
  numeric literals, rules 17-6
mode indicators
  AC 3-12
  CG 3-12
  DL 3-12
  IN 3-12
  IP 3-12
  IS 3-12
  KG-KL, KQ-KY 3-12
  PG 3-12
  RP 3-12
  RR 3-12
  RU 3-12
  RV 3-12
  01-89 3-12
mode-level fields
  display station fields 2-23
  on generation output 9-10
  on m-specification 15-9
modes
  changing 10-2
  delete 2-4
  enter 2-4
  insert 2-4
  operating 10-2
  review 2-4
  when you start 10-2
modes of operation
  changing 10-2

  delete 2-4
  enter 2-4
  insert 2-4
  operating 10-2
  review 2-4
  when you start 10-2
modulus 10 A-1
  calculations A-1
modulus 10 self-check field 16-21
modulus 11 A-2
  calculations A-2
modulus 11 self-check field 16-21
MOVE
  explanation 7-30
  requirements 18-16
move array elements 18-17
move operations
  MOVE 18-16
MOVEA
  explanation 7-13
  requirements 18-17
MOVEL
  explanation 7-30
  requirements 18-18
  summary of rules 18-18
MRT procedure 8-18
MRT program 2-23
MRTMAX 8-19
MSG
  in C-specifications 7-27
  in preprocessing 7-10
  in subroutines 7-7
  uses 18-20
MULT
  explanation 7-11
  requirements 18-22
MVR
  explanation 7-11
  requirements 18-23

N

name field information 9-10
negative value, how to enter 10-10
NEP
  on WSU generated procedure 8-21
Network Resource Directory (NRD) B-3
never-ending program
  on WSU generated procedure 8-21
non-contiguous keys 4-1, 4-11, 12-3
non-null characters 16-20
normal and expected data from testing 9-14
not-found indicator
  turned off 15-3
  turned on 15-3
NRD (Network Resource Directory) B-3
Number of Lines to Clear 16-3
numeric data to alphameric data 18-16, 18-18
numeric field
  binary 12-7
  packed decimal 12-7
  processing 16-19
  zoned decimal 12-7
numeric output fields 16-17

O

operating modes 10-2
operation codes
  chart of 17-8
  list 7-4
operations 17-6
  alphabetic 12-6
  arithmetic
    ADD, explanation 7-11
    ADD, requirements 18-1
    DIV, explanation 7-11
    DIV, requirements 18-4
    length of 7-11
    list of 7-4
    MULT, explanation 7-11
    MULT, requirements 18-22
    MVR, explanation 7-11
    MVR, requirements 18-23
    SUB, explanation 7-11
    SUB, requirements 18-29
    Z-ADD, explanation 7-11
    Z-ADD, requirements 18-30
    Z-SUB, explanation 7-11
    Z-SUB, requirements 18-30
  array
    list of 7-4
    LOKUP, explanation 7-13
    LOKUP, requirements 18-14
    MOVEA, explanation 7-13
    MOVEA, requirements 18-17
    SORTA, explanation 7-13
    SORTA, requirements 18-28
    XFOOT, explanation 7-13
    XFOOT, requirements 18-30
  branching
    GOTO, explanation 7-19
    GOTO, requirements 18-12
    list of 7-4
    TAG, explanation 7-19
    TAG, requirements 18-29
  comparing and testing
    COMP, explanation 7-20
    COMP, requirements 18-2
    list of 7-5
    RANGE, comparing and testing 7-20
    RANGE, explanation 7-24
    RANGE, requirements 18-27
  conditioning of 2-17
  debugging
    list of 7-5
    requirements 7-24, 18-2
  functions 7-11
  indicator setting on and off
    list of 7-5
    SETOF, explanation 7-25
    SETOF, requirements 18-27
    SETON, explanation 7-25
    SETON, requirements 18-28
  input and output
    GET, explanation 7-25
    GET, requirements 18-6
    GETNH, explanation 7-25
    GETNH, requirements 18-7
    GETNR, explanation 7-25
    GETNR, requirements 18-8
    GETPH, explanation 7-25
    GETPH, requirements 18-10
    GETPR, explanation 7-26
    GETPR, requirements 18-11
    IMSG, explanation 7-26
    list of 7-5
    MSG, explanation 7-27
    MSG, requirements 18-20
    PUT, explanation 7-28
    PUT, requirements 18-24
    PUTN, explanation 7-29
    PUTN, requirements 18-24
    PUTS, requirements 18-25
    TIME, explanation 7-29
  move
    list of 7-5
    MOVE, explanation 7-30
    MOVE, requirements 18-16
    MOVEA, explanation 7-13
    MOVEA, requirements 18-17
    MOVEL, explanation 7-30
    MOVEL, requirements 18-18
  numeric 12-8
  results 17-9

- subroutine
    - BEGSR, explanation 7-33
    - BEGSR, requirements 18-1
    - ENDSR, explanation 7-33
    - ENDSR, requirements 18-4
    - EXSR, explanation 7-33
    - EXSR, requirements 18-5
    - list of 7-5
    - PRTY, explanation 7-33
    - PRTY, requirements 18-23
- option
  - option 8-18
  - option on WSX statement 8-22
- OR relationships 4-15
- output
  - where stored 13-3
- output data, restrictions 16-17
- output fields
  - constants or prompts 6-9
  - D-specifications 6-9
  - numeric format 13-4
- output library name 9-13
- output/input errors
  - GET 18-6
  - GETNR 18-8
  - GETPH 18-10
  - GETPR 18-11, 18-12
- output/input fields 6-9
- overlaying segments of program B-1

P

- packed decimal format
  - data format 4-18
  - on I-specification 12-7
- packed fields, length 4-19
- page backward 2-13
- page backward group 10-4, 15-3
- page forward 2-13
- page forward group 10-4, 15-3
- physically deleting records B-3
- position of data within a line 16-16
- preprocessing
  - explanation 7-9
  - on C-specification 16-11
- preventing unpredictable results 6-4
- priority
  - assigning B-1
  - of display frequency 16-10
  - restrictions 16-10
- procedures
  - WSUTXCR 19-7
  - WSUTXEX 19-1
  - WSUTXRV 19-12
- Processing (C) Specifications
  - comments 7-6, 17-14
  - conditioning indicators 17-3
  - Decimal Positions 7-5, 17-10
  - Factor 1 7-4, 17-5
  - Factor 2 7-4, 17-5
  - Half Adjust 7-5
  - half-adjust 17-11
  - length 17-9
  - literals 17-6
  - name 17-9
  - operation 17-6
  - Processing Function 7-3, 17-2
  - RANGE operation 17-6
  - Result Field 7-5, 17-9
  - Result Field Length 7-5
  - Resulting Indicators 7-6
    - plus or high 17-12
    - zero or equal 1=2 17-12
  - table or message text 17-6
- Processing Function Indicators
  - AN or OR 7-3, 17-2
  - EJ 7-3
  - ES 7-3, 17-2
  - EW 7-3, 17-2
  - IJ 7-3, 17-2
  - IW 7-3, 17-2
  - SR 7-3, 17-2
- processing level indicator
  - list 17-2, 17-4
- processing levels
  - figure 2-6
  - in enter mode 2-5
- processing, interruption 18-25
- program
  - testing 9-14
  - WSU source 1-9
- program cycle 2-4
- program date
  - date, format 13-4
  - format 13-4
- program mode indicator
  - on c-specification 17-3
  - set off 3-23
  - set on 3-23
  - when used 17-3
  - where coded 3-23
- program name 13-1
- programmable indicators
  - AC 3-8
  - AE 3-8
  - EJ 3-8
  - ES 3-8
  - EW 3-8
  - IS 3-8
  - JA-JN, JP-JY 3-8
  - KG-KL, KQ-KY 3-8
  - PG 3-8
  - RC 3-8
  - RP 3-8
  - RS 3-8
  - SA-SN, SP-SY 3-8
  - U1-U8 3-8
  - 01-89 3-8
- programmed halts, displayed text for 9-11
- programs, example
  - SE100W
    - coding sheets 11-55
    - explanation 11-54
    - printout 11-60
  - 1M110W
    - coding sheets 11-5
    - explanation 11-1
    - printout 11-17
  - 1M300W
    - coding sheets 11-33
    - explanation 11-30
    - printout 11-42
- prompt
  - coding 16-30
  - prompt 6-10
  - SEU 8-5
  - uses of 6-10
- prompt display
  - WSUTXCR 19-7
  - WSUTXEX 19-2
  - WSUTXRV 19-12
- protect field, restrictions 16-23
- PRTY
  - explanation 7-33
  - requirements 18-23
- PUT
  - explanation 7-28
  - requirements 18-24
- PUTN
  - explanation 7-29
  - requirements 18-24
- PUTS operation
  - explanation 7-29
  - in preprocessing 7-9
  - in review mode 2-17
  - in special processing 7-34
  - operations 16-2

    uses of 18-25

R

RANGE
    explanation 7-20
    requirements 18-27
record
    recover inserted 19-7
    unchained B-4
record chains
    creating from several records 4-6
    excluding or reordering 4-6
record deletion 1-6
record deletion, prevention B-4
record identification character
    copying records 19-1
    create WSU file from non-WSU 19-1
    digit 12-6
    digit portion 12-6
    maximum per line 4-14
    portion used 12-6
    remove blank records 19-1
    removing records 19-12
    special 12-5
    zoned portion 12-6
record insertion 1-6
record insertion, how to 10-13
record key
    alphameric 12-3
    length 12-3
    packed decimal 12-3
    starting location 12-3
record lines
    I-specification 4-12
    include 4-12
    maximum number 4-12
record review 1-6
record review type 16-11
record type
    indicators 12-5
    relationship 12-4
record-not-found condition
    GETNH 18-7
    GETNR 18-8
    GETPH 18-9
    GETPR 18-11
recovery-of-work session indicator
    set off 3-23
    set on 3-23
    when used 17-4
    where coded 3-23
region size
    modify on J-specification 13-3
    on J-specification 13-3
    WSU-generated procedure 8-18
relative record number 2-14
    adding or updating 10-7
    print 4-6
remote files 1-7, 4-1, 4-2, B-3
repeat display 16-10
Repeat, restrictions 16-10
repeated-display indicator
    set off 3-24
    set on 3-24
    when used 17-3
    where coded 3-24
reports printed 1-6
reserved fields
    list 3-1
Reset Keyboard, restrictions 16-5
restrictions 20-2
result field
    decimal positions 17-10
    define 17-9
    half-adjust 17-11
    packed fields 17-9
resulting indicators
    additional uses 17-14
    before an operation 17-12
    IMSG operation 17-14, 18-13
    LOKUP operation 18-14
    MSG operation 17-14
    PUT operation 17-14
    PUTN operation 17-14
    SETOF operation 17-14
    SETON operation 17-14
    turning on 17-12
resume enter mode 10-4
resume review 2-15
resume review mode 10-4
resume-work-session indicator
    set off 3-26
    set on 3-26
    when used 17-4
    where coded 3-26
return fields to program 16-4
Return Input, restrictions 16-4
reverse data image 16-27
review mode 2-4
    Bypass Display command key 2-12
    combined enter/review displays 2-11
    display sequences 2-12
    displays 2-11
    functions 2-13
        page backward 2-13
        page forward 2-13
    how to get 2-9
    how to select 10-12
    indicator 3-26
        set off 3-26
        set on 3-26
        where coded 3-26
    mode of operation 2-4
    relative record number 2-14
    resume review 2-15
    review only 2-10
    review with master file update 2-10
    review with update 2-10
    separate review mode displays 2-11
    variations 2-10
    when allowed 2-11
    WSU's role 2-16
        issue message 2-16
        place relative record number 2-16
        retrieve records 2-16
        save fields 2-16
        search S-specifications 2-16
        set mode-level fields 2-16
        set on RV and 99 2-16
        show requested record 2-16
review mode displays 2-11
Review Mode Record Identifying Indicators, restrictions 16-11
review only mode 2-10
review with master file update 2-10
review with update 2-10
review-capable 2-9
review-mode indicator
    review mode indicators 16-11
    set off 3-26
    set on 3-26
    where coded 3-26
review-record indicator
    on c-specification 17-3
    set off 3-25
    set on 3-25
    when used 17-3
    where coded 3-25
right adjust field 16-21
Roll Down key
    enabled 16-6
    Key Mask 16-13
Roll Up key
    enabled 16-6
    Key Mask 16-13
rover-update indicator
    set off 3-26

  - set on 3-26
  - where coded 3-26
- RPG F-(File Description), entries
  - File name 12-2
  - File Type 12-2
  - Key Field Starting Location 12-3
  - Length of Key Field 12-3
  - Record Address Type 12-3
  - Record Length 12-2
  - Type of File Organization 12-3
- RV indicator 2-4

S

- S-specification
  - blink cursor 6-4, 16-8
  - display format name 6-3, 16-1
  - enable command keys 6-4, 16-7
  - enable function keys 6-4, 16-6
  - enter-mode sequence
    - End 16-9
    - enter-mode sequence 6-5
    - Entry Required 16-10
    - Repeat 16-10
    - Start 16-9
  - format ID 6-3, 16-2
  - insert mode indicators 6-8
  - key mask 6-4, 16-12
  - lowercase 6-4, 16-4
  - Null Fill 16-9
  - Number of Lines to Clear 16-3
  - preprocess 6-7, 16-11
  - priority 6-7, 16-10
  - Reset Keyboard 16-5
  - return input 6-4, 16-4
  - review mode indicators 6-8
  - Review Mode Record Identifying Indicator 16-11
  - Review Mode Record Identifying Indicators 16-11
  - sound alarm 6-4, 16-5
  - Start Line Number 16-3
- S-specifications
  - location 16-1
  - required entries 16-1
- SCR information 9-10
- screen design aid
  - use in WSU 6-1
- SDA
  - use in WSU 6-1
- SDAD1 8-8
- SDAD2 8-8
- search word, LOKUP operation 18-14
- selection of desired display 16-2
- selection of displays by ID 16-2
- selections of WSU display 16-2
- self-check digit, formulas A-1
- self-check field
  - calculating A-1
  - how to code 16-21
  - numeric A-1
  - verification A-1
- sequence, arrays 14-2
- session field headings 9-10
- session indicators
  - AE 3-10
  - ES 3-10
  - EW 3-10
  - indicators 3-27
    - set off 3-27
    - set on 3-27
    - where coded 3-27
  - IW 3-10
  - RC 3-10
  - RS 3-10
  - SA-SN, SP-SY 3-10
  - U1-U8 3-10
- session-level fields
  - in file definition 15-6
  - on m-specification 15-6, 15-9
  - values of 2-23
  - where defined 2-23
- session-level fields, initialization 4-29
  - used in the program 9-10
- session-to-session indicators 3-27
- SETOF 7-25
- SETOF, uses 18-27
- SETON 7-25
- SETON, uses 18-28
- SEU 8-1
- SEU procedure prompt 8-2, 8-5
  - illustration 8-2
  - specification display types 8-3
- sign in numeric data 4-16
- sign, on numeric data 4-20
- simple data entry 1-5
- SORTA
  - explanation 7-13
  - requirements 18-28
- source entry utility, using 8-1
- source members, multiple 8-4
- source program 8-5
  - entering 8-5
  - restriction 8-8
- source program information
  - error message numbers 9-6
  - source statements and file 9-6
  - statement numbers 9-6
- space 20-1
- special WSU fields 3-1
- specification display types 8-7
- specification order 8-6
- specifications, entering and updating 8-1
- specifications, order for WSU programs 1-2
- SR, when used 17-2
- Start Line Number, equation 16-3
- Start, restrictions 16-9
- starting line number 3-3
- statement number
  - in source program information 9-6
- STMT# information 9-10
- storage requirements 9-12
- SUB
  - explanation 7-11
- subroutine operations
  - BEGSR 7-7
  - BEGSR (begin subroutine) 18-1
  - ENDSR 7-7
  - ENDSR (end subroutine) 18-4
  - EXSR (execute subroutine) 18-5
  - GOTO 7-7
  - PRTY (priority) 18-23
  - TAG 7-7, 18-29
- subroutines
  - beginning 18-1
  - branching point 18-4
  - ending 18-4
  - EXSR 7-7
  - MSG operation 7-7
  - requirements 7-7
  - subroutines 7-7
  - using PRTY 7-7
  - where coded 7-33
- SUBR22 19-1
- symbolic display station identifier 3-6
- system date 7-29
- system time 7-29
- system time-of-day 18-30
- System/36 Display Format Specifications coding sheet 16-1

T

T-specification
  coding 4-27
  error indicator 15-4
  file definition location 15-1
  file name 15-1
  header record identifying indicator, use 15-1
  library name 15-1
  location 15-1
  not-found indicator 15-1
  number of records 15-1
  number per program 4-25
  number per WSU program 4-27
  required entries 4-27
  source member name 15-1
  use of 4-27
T-specification entries
  error indicator 4-28
  file name 4-27
  header record identifying indicator 4-27
  library name 4-27
  not-found indicator 4-27
  number of records 4-28
  source member name 4-27
table or message text, IMSG operation 18-13
TAG
  explanation 7-19
  requirements 18-29
template, work station 10-3
test data 9-14
testing, how to shorten 9-16
TIME
  explanation 7-29
  requirements 18-30
total fields
  accumulating totals B-2
  restoring B-2
  saving B-2
totals, back out 7-9
trailers
  chart 4-9
  data record 4-9
  job control record 4-9
  symbolic identifier 4-8
  transaction files 4-8
  work session control record 4-9
transaction file
  *RLNO 4-3
  concatenation 4-6
  control information 1-7
  creation 19-7
  creation from non-WSU file 4-6
  data from 1-7
  deleting records in 10-14
  description 4-3
  detail records 1-7
  error codes 3-4
  explanation 4-2
  header records 1-7
  inserting records into 10-13
  job control record 4-3
  logical organization 4-8
    figure 4-8
  maximum entry 12-7
  maximum length 12-5
  maximum number 1-7
  modes 2-4
  MOVEA 18-17
  name 8-18
  non-WSU program 4-8
  not-found indicators 18-8
  number of records 8-18
  number per WSU program 4-27
  output/input errors
    GETPH 18-10
    GETPR 18-11, 18-12
  prepare for use by non-WSU program 4-5
  reading records 4-3
  record length 12-2
  record-not-found condition
    GETNH 18-7
    GETNR 18-8
    GETPH 18-9, 18-10
    GETPR 18-11
  recovering 4-4
  recovery 4-7
  remote files 1-7, 4-2, B-3
  remove blank records 4-6
  remove or reclaim partially inserted records 4-7
  remove partially inserted records 4-7
  removing trailers 4-4
  review a record 4-6
  review records in 10-12
  unique 4-2
  use of 1-4
  work session identifier 4-8
  writing records 4-3
two's complement 4-20

U

underline data 16-27
universal work session selection 8-18
  UNIV-ALL 8-22
  UNIV-id 8-22
unpredictable results, preventing 6-4
updating master file records 2-17
user command keys 10-5
user message member MICs 9-9
user-defined help displays B-5
using ideographic characters 20-1
using this manual 1-1
  as a guide 1-1
  as a reference 1-3
  examples 1-3

V

variable data field 6-9
variable line numbers 16-16
variable starting line number 16-3

W

work
  definition 2-1
  ending 10-6
  ending a 10-14
  explanation 10-2
  identifier 4-8
  initiation 17-3
  on c-specification 17-4
  restarting 10-6
  when used 17-4
work session
  definition 2-1
  ending 10-6
  ending a 10-14
  explanation 10-2
  identifier 4-8
  initiation 17-3
  on c-specification 17-4
  restarting 10-6
  when used 17-4
work session chains, extraction of 19-1
work-session-initiation indicator
  set off 3-21
  set on 3-21
  where coded 3-21
WSU
  create procedure 19-7
  extract procedure 19-1
  recover procedure 19-12

WSU aid display 10-8
WSU aid display, how to get 10-8
WSU arrays, on d-specification 16-15
WSU command keys
  accept sequence error 10-4
  bypass display 10-4
  cmd 1 WSU display 10-4
  cmd 13 accept sequence error 10-4
  cmd 14 delete mode 10-4
  cmd 15 resume mode 10-4
  cmd 2 bypass display 10-4
  cmd 5 page backward group 10-4
  cmd 6 page forward group 10-4
  delete mode 10-4
  insert mode 10-4
  Key Mask 16-13
  page backward group 10-4
  page forward group 10-4
  resume enter mode 10-4
  WSU display 10-4
WSU display
  cmd key 1 10-4
WSU display, contents
  add/update relative record 10-7
  chain-end indicator field 10-7
  display station identifier 10-6
  end work session 10-6
  format/level id field 10-7
  mode/processing level field 10-7
  review record number 10-6
  session id field 10-7
  session selection identifier 10-6
WSU field, starting location 16-16
WSU files 4-1
WSU J-specification
  coding 4-26
  location 13-1
  number per program 4-25
  required entries 4-26
WSU job
  explanation 10-2
  how WSU works 2-1
WSU job, array, and file specifications 4-25
WSU M-specification
  coding 4-29
  error indicator 15-7
  field level 15-9
  file definition location 15-6
  file definition type 15-8
  file name 15-5
  get field names 15-7
  library name 15-6
  location 15-5
  not-found indicators 15-7
  number per program 4-25
  source member name 15-6
  use of 4-29
WSU options 9-5
WSU procedure command 1-9
  generating your program 8-9
  order 8-15
  parameters 8-15
WSU procedure display, default values 8-10
WSU procedure prompt, defaults 9-5
WSU program
  comments on 17-14
  how to generate 8-9
  how to run 10-1
  name 8-18
  order of specifications 1-2
  place on JOBQ 8-14
  starting 10-9
  steps for writing 1-2
    code 1-2
    enter 1-2
    generate 1-2
    run 1-3
    test and debug 1-3
  what WSU produces 1-10
  what you should know 10-1
WSU prompt 8-9
  modify on WSU-generated procedure 8-21
WSU prompt information
  Cmd 4-put on JOBQ 8-14
  cross-reference level 8-13
  generation option
    NOPGM 8-12
    NOPROC 8-12
    PGM 8-12
    PROC 8-12
  name of library 8-10
  name of source program 8-10
  save display format source 8-14
  size of work file in blocks 8-12
  source print option
    LIST 8-11
    NOLIST 8-11
    NOLISTS 8-11
    NOLISTW 8-11
  stop option
    HALT 8-11
    NOHALT 8-11
    NOSTOP 8-11
    REPLACE 8-11
WSU prompt, defaults 8-10
WSU T-specification
  coding 4-27
  error indicator 15-4
  file definition location 15-1
  file name 15-1
  header record identifying indicator, use 15-1
  library name 15-1
  location 15-1
  not-found indicator 15-1
  number of records 15-1
  number per program 4-25
  number per WSU program 4-27
  required entries 4-27
  source member name 15-1
  use of 4-27
WSU-C 8-8
WSU-D 8-8
WSU-E 8-7
WSU-generated procedure
  information contained on
    debugging option 8-22
    display formats load member library name 8-18
    load member containing display formats 8-18
    load member name 8-18
    master file name 8-18
    maximum number of display stations 8-18
    message member library name 8-18
    number of records 8-18
    region size 8-18
    transaction file name 8-18
    universal work session selection 8-22
    WSU program library name 8-18
    WSU program name 8-18
  modifying 8-19
WSU-J 8-7
WSU-M 8-7
WSU-T 8-7
WSU's role in review mode 2-16
WSUTXCR
  command 19-7, 19-10, 19-11
  parameters 19-7
    create option 19-8
    diskette location 19-9
    length of record 19-8
    name of input file 19-8
    name of library 19-9
    name of output file 19-8
    name of program 19-9
    session identifier 19-8
    size of file in records 19-8
    type of file 19-8
  procedure 19-7
  prompt display 19-7

WSUTXEX
  command 19-5
  examples 19-6
  functions 19-1
  parameters
    copy option 19-2
    length of records 19-4
    location of diskette 19-4
    name of input file 19-2
    number of records for output 19-4
    output option 19-3
    retention days or ADD 19-4
    session identifier 19-3
    type of file 19-2
    volume ID of diskette 19-4
  procedure 19-1
  prompt display 19-2
WSUTXRV
  functions 19-12
  parameters 19-13
    command 19-14
    name of library 19-13
    name of program 19-13
    name of transaction file 19-13
    record processing option 19-13, 19-14
  procedures 19-12
  prompt display 19-12

X

XFOOT
  explanation 7-13
  requirements 18-30

Z

Z-ADD
  explanation 7-11
Z-SUB
  explanation 7-11
  MVR (move remainder) 18-22
  requirements 18-30
zoned decimal format
  data format explanation 4-16
  P/B/L/R on I-specification 12-7

Numerics

01-89 indicators
  set off 3-28
  set on 3-28
  where coded 3-28
5292 color display station
  blink field 16-25
  chart 16-26
  column separators 16-28

IBM System/36:
Work Station Utility Guide

SC21-7905-2

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name ____________________

Company or
Organization ____________________

Address ____________________

____
City State Zip Code

No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold and tape　　Please do not staple　　Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS　　PERMIT NO. 40　　ARMONK, N. Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM CORPORATION
Information Development
Department 532
Rochester, Minnesota, U.S.A. 55901

Fold and tape　　Please do not staple　　Fold and tape

IBM

IBM

Work Station Utility Guide

International Business Machines Corporation

Contents

1 Introduction
2 How WSU Works
3 Reserved Fields and Indicators
4 Coding Files
5 Coding Arrays
6 Coding Displays
7 Coding Processing
8 Entering and Generating a WSU Program
9 Testing and Debugging WSU Programs
10 ATTENTION: OPERATORS (Running a WSU Program)
11 WSU Example Programs
12 F-(File Description) and I-(Input) Specifications
13 J-(Job) Specification
14 E-(Array) Specifications
15 T-(Transaction File) and M-(Master File) Specifications
16 S-(Display Control) and D-(Field Definition) Specifications
17 C-(Processing) Specification
18 Operation Codes
19 File Processing Procedures
Appendixes
Glossary
Index

File Number
S36-32

Order Number
SC21-7905-2

Printed in U.S.A.

SC21-7905-02