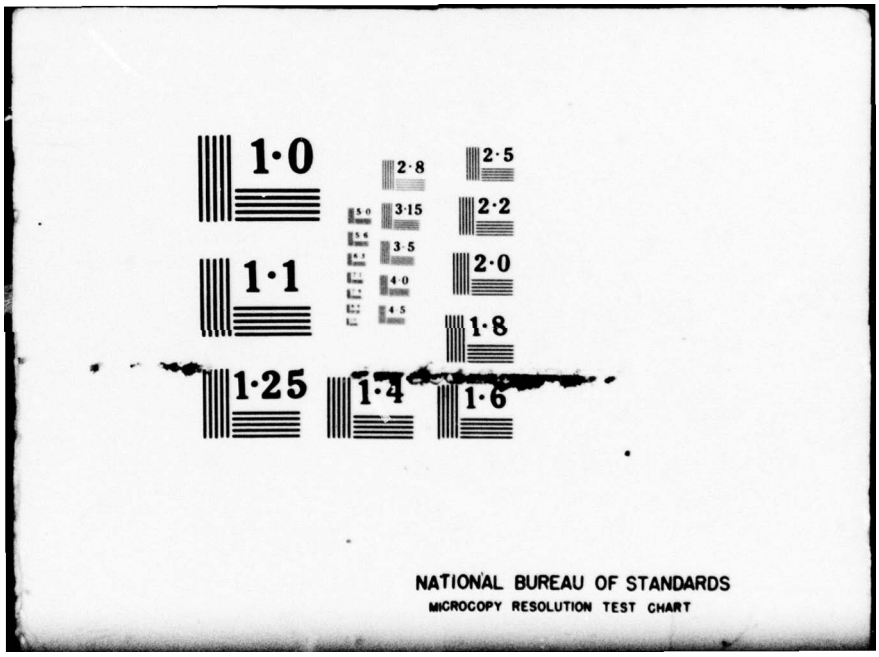
AD-A048 478 PRC DATA SERVICES CO MCLEAN VA F/G 9/2
WORLD WIDE DATA MANAGEMENT SYSTEM (WWDMS) SELF INSTRUCTION MANU--ETC(U)
SEP 77 R A KOGAN DCA100-73-C-0015
UNCLASSIFIED PRC/DSC-606 NL

1 OF 1
AD A048478



END
DATE FILMED
2-78
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AD A048478

CCTC

4 NW



COMMAND & CONTROL TECHNICAL CENTER

DDC
JAN 5 1978
F

WORLD WIDE DATA MANAGEMENT SYSTEM (WWDMS)

SELF INSTRUCTION MANUAL T-2 VERSION

ADDENDUM B

30 SEPTEMBER 1977

AD NO.
DDC FILE COPY

DEFENSE COMMUNICATIONS AGENCY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DEFENSE COMMUNICATIONS AGENCY
COMMAND AND CONTROL
TECHNICAL CENTER
WASHINGTON, D. C. 20301

3 JAN 1978

IN REPLY
REFER TO: C422

TO: Director
Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

SUBJECT: Forwarding of Worldwide Data Management System (WWDMS) Self Instruction Manual -- T-2 Version (Addendum B)

Reference:
(a) DCA Ltr, C422, WWDMS Self Instruction Manual -- T-2 Version; forwarding of, 18 Aug 76
(b) DCA Ltr, C422, WWDMS Self Instruction Manual T-2 Version (Addendum A); forwarding of, 26 Oct 76

1. Four copies of subject addendum dated 30 September 1977 are forwarded herewith for processing by your organization. This addendum was cleared for open publication on 12 Dec 77 by the Directorate for Freedom of Information and Security review (OASD-PA), Department of Defense.

2. In addition, we are enclosing a Report Documentation Page (DD Form 1473) and a DDC Accession Notice (DDC Form 50) pertaining to this manual.

3. Please note that this is an addendum to the WWDMS Self Instruction Manual -- T-2 Version dated 26 July 1976 as forwarded to you by reference (a). The Accession Number for the basic manual is AD-A028813. Addendum A was forwarded to you by reference (b) and was assigned Accession Number AD-A031655. These relationships should be preserved.

FOR THE DIRECTOR:

3 Enclosures a/s

BIRTRUN S. KIDWELL, JR.
Colonel, USA
Deputy Director for WWMCCS ADP

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: N/A	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Worldwide Data Management System (WWDMS) Self Instruction Manual -- T-2 Version (Addendum B)	5. TYPE OF REPORT & PERIOD COVERED: FINAL Dec 1975 - Sep 1976
6. PERFORMING ORG. REPORT NUMBER: PRC Task #606	
7. AUTHOR(s): Ronald A. Kogan, et.al.	8. CONTRACT OR GRANT NUMBER(s): DCA100-73-C-0015
9. PERFORMING ORGANIZATION NAME AND ADDRESS: PRC Data Services Company, 7600 Old Springhouse Road, McLean, VA 22101	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: PE. 32017K, PROJECT: 420300, TASK: 420303
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Communications Agency, Command and Control Technical Center (C422), Pentagon, Washington, D.C.	12. REPORT DATE: 30 Sep 77
	13. NUMBER OF PAGES: 63
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same as Block 11	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution Unlimited

DDC
RECEIVED
JAN 5 1978
F

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

The basic manual is based upon software obtained by the U.S. Government from Honeywell Information Systems Inc. in support of the Worldwide Military Command and Control System (WWMCCS). This report covers the second set of changes (Addendum B) to the manual.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

WWDMS; Worldwide Data Management System; WWMCCS; Data Management Systems; Self Instruction Manual; Data Base Management Systems; DBMS; Honeywell 6000 Series Computer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of the basic self instruction manual is to allow a person without prior experience in programming languages or data management systems to learn to use the Worldwide Data Management System (WWDMS) as implemented on the Honeywell Series 6000 computer. The first section consists of elementary concepts and represents a stand-alone elementary user's self instruction manual. The second section contains advanced concepts, and when used in conjunction with the first section, comprises a complete advanced WWDMS user's self instruction manual.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



C
C
T
C



(4)
(10) Ronald A. Kogan

(9) Final rept.
Dec 75 - Sep 76

COMMAND & CONTROL TECHNICAL CENTER

(15) DCA100-73-C-0015 (14) PRC/DSC-606

(6)

WORLD WIDE DATA MANAGEMENT SYSTEM (WWDMS)

SELF INSTRUCTION MANUAL T-2 VERSION.

ADDENDUM B.

(11) 30 SEPTEMBER 1977

(12) 68p.

DEFENSE COMMUNICATIONS AGENCY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

1977

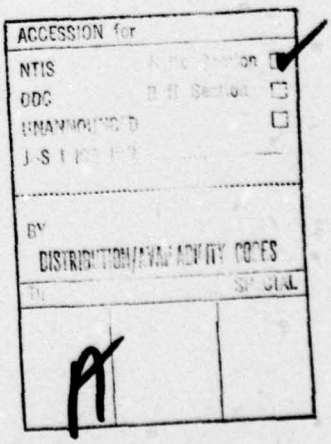
408 424

1B

COLLATING INSTRUCTIONS

To incorporate Addendum B changes into the T-2 SIM, remove old pages and replace with corresponding new pages as indicated below.

REMOVE	REPLACE WITH
Title Page	Title Page
-	Record of changes
xiii/xv	xiii/xv
20-1	20-1/20-42
I-1/I-16	I-1/I-17



CONTENTS (Continued)

Lesson Number		Page
20	QUERY PROCEDURES	20-1
	Default APPLICATION DEFINITION	20-2
	QUERY Statement	20-8
	Conditional Expression	20-13
	PRINT Statement	20-15
	END Statement	20-19
	REPORT Statement	20-20
	PRINT Label Statement	20-21
	PRINT Report-Name Statement	20-21
	LINE Statement	20-23
	SPACE Statement	20-25
	SORT Statement	20-25
	SORT ON... Statement	20-32
	SORT Report-Name ON... Statement	20-32
	SORT Identifier ON... Statement	20-33
	COPY Statement	20-33
	LET Statement	20-34
	LIBRARY Statement	20-36
	OPTIONS ARE Statement	20-37
21	DIRECT ACCESS PROCEDURES	21-1
	RUNP Command	21-1
	WWDMS Terminal Executive Batch Processor (DMTEX)	21-2
	PRINT/REPORT ON DAC Options	21-3
	ACCEPT Statement	21-4
	DISPLAY Statement	21-6

CONTENTS (Continued)

Lesson Number		Page
22	MISCELLANEOUS PROCEDURE CONCEPTS	22-1
	Parameterized Procedures	22-1
	Source Parameters	22-1
	Object Parameters	22-6
	Run-time Options	22-9
	RUN Command Options	22-9
	OPTIONS ARE Statement	22-14
	COPY Statement	22-15
	KEEP/RETRIEVE...SUBSET STATEMENTS	22-16
	HOLD Statement	22-18
	DIRECT Statement	22-20
23	DATA FILE CREATION	23-1
	CREATE Statement	23-1
	PLACE Clause	23-2
	EQUATE Clause	23-3
	INHIBIT Clause	23-3
	Transaction Reject File	23-5
	Create Errors	23-6
	Out-of-sequence Errors	23-6
	Check Errors	23-6
	Implicit CREATE Procedures	23-11
	STORE Statement	23-13
	REJECT Statement	23-14
	WHEN...TRANSACTION SEQUENCE ERROR Statement	23-14
	WHEN...CHECK ERROR Statement	23-15
	WHEN...REJECT ERROR Statement	23-16
	AUDIT	23-17
	TEST Option	23-18
24	DATA FILE MAINTENANCE	24-1
	Advantages and Disadvantages to WWDMS Procedure Maintenance	24-1
	UPDATE Statement	24-2
	MATCH Clause	24-4
	WHERE/WITH Clause	24-4
	EQUATE Clause	24-4
	INHIBIT Clause	24-5
	AUDIT Clause	24-5
	Transaction Reject File	24-9
	Implicit UPDATE Procedures	24-10
	UPDATE Errors	24-11
	Explicit UPDATE Procedures	24-13
	WHEN...ENTRY Statement	24-14

CONTENTS (Continued)

Lesson Number		Page
	WHEN...MATCHED Statement	24-14
	WHEN...UNMATCHED TRANSACTION Statement	24-14
	WHEN...UNMATCHED MASTER Statement	24-14
	WHEN...END OF TRANSACTION FILE Statement	24-15
	WHEN...END OF MASTER FILE Statement	24-15
	WHEN...TRANSACTION SEQUENCE ERROR Statement	24-15
	WHEN...MASTER SEQUENCE ERROR Statement	24-15
	WHEN...CHECK ERROR Statement	24-15
	WHEN...REJECT ERROR Statement	24-15
	DELETE Statement	24-16
	INSERT Statement	24-16
	RESTORE Statement	24-17
	STORE Statement	24-17
	REJECT Statement	24-17
	TEST Option	24-20
	UPDATE...CREATING Statement	24-21
	RETRIEVE...FOR UPDATE Statement	24-24
	IDS File Maintenance	24-30
	Record Deletion	24-30
	Data Field Modification	24-33
	Chain Linkage Modification	24-33
	Control Field Modification	24-35
	New Record Insertion	24-36
	WWDMS Maintenance Capabilities	24-37
25	WWDMS SUPPORT SUBSYSTEMS	25-1
	WWTUT	25-1
	DFAULT	25-4
	WWLI	25-5
	WWDIRUPD	25-7
	PERFORM	25-9

APPENDIXES

A.	WWDMS BASIC COMMAND LANGUAGE (WWBCL)	A-1
B.	WWDMS INTERFACE SOFTWARE PACKAGE (WISP)	B-1
C.	DATA FILE STRUCTURES	C-1
	Sequential Files	C-1
	Indexed-Sequential Processor (ISP) Files	C-3
	Inverted Files	C-7
	Integrated Data Store (IDS) Files	C-8

CONTENTS (Continued)

Lesson Number		Page
D.	USER ROUTINE LIBRARY CREATION	D-1
	System Loadable Files	D-1
	User Routine Library Structure	D-1
	User Subroutines/Functions	D-7
	PERFORM subsystem LIBRARY program	D-11
	User Table Lookup Routines	D-25
	PERFORM subsystem WWTLU program	D-30
	WWLI	D-33
I.	INDEX	I-1

LESSON 20
QUERY PROCEDURES

20-1. The QUERY Procedure Language is a subset of the WWDMS Procedure Language. It is a simple set of statements that allows a user to create a detailed output report without having to write a complex procedure to retrieve any necessary data items. Merely specifying data items that will be used in the generation of an output report will result in automatic retrieval of any required logical records from a data base. The concept of entries containing selected data elements from a number of logical records does not apply to a QUERY procedure.

20-2. There are a number of conceptual differences between QUERY and WWDMS procedures. They are as follows:

- Entry names are not valid within the syntax of the QUERY Procedure Language. The QUERY statement specifies an Application Definition File (ADF) in its syntax. A QUERY procedure containing data items that comprise an output report will cause the specified ADF to be searched in an attempt to associate the requested data items with existing entries. These entries will be automatically retrieved in an operation that is transparent to the user.

- The data items specified in a QUERY procedure must be contained in the same data base. Any COORDINATED ENTRIES in an ADF may not be used in a QUERY procedure.

- Items referenced in a QUERY procedure cannot be subscripted.

- The only labels allowed within a QUERY procedure are report-packet-labels (LINE and SPACE statement labels). Event labels, Routine labels, and Procedure Control labels are all illegal.

- The only statements allowed in a QUERY procedure are:

COPY	LINE	REMARK
END	OPTIONS ARE	REPORT
LET	PRINT	SORT
LIBRARY	QUERY	SPACE

  These statements, although functionally similar to their WWDMS procedure counterparts, may be syntactically unique in a QUERY procedure.

20-3. It is appropriate to introduce a new concept at this time. In Lessons 8 and 9 you were introduced to the Data Definition File, the Application Definition File, and the relationship between entries in the ADF and logical records in the DDF. It was shown how a number of entries could be defined, each containing a number of combinations of logical records.

In Frame 20-2, you were told that a QUERY procedure does not reference entries, but data elements. The WWDMS processor automatically matches these data items to appropriate entries in the ADF referenced by a QUERY procedure. A problem may arise if many entries exist that contain the same logical record. WWDMS will not know which entry to use to retrieve a required logical record.

20-4. With certain kinds of data base structures, this problem can be easily avoided by using a Default Application Definition File in your QUERY procedure reference. A Default ADF is created by the Data Base Administrator using a special batch routine. This routine is given a Directory File as input, from which it extracts information contained in the associated object DDF. It creates a source ADF from the DDF, and then passes this source to the normal ADFXLT program for a source-to-object translation.

The Default ADF is created using the following rules:

1. An entry is created for every record in the DDF. Each of these entries contains one and only one logical record. The entry-name is the same as the logical record name.

   For example, the logical record

```
01 PERSON-REC
   ;TYPE IS "A" IN TYPE-FIELD
   ;RETRIEVAL VIA SCAN.
```

results in an entry

```
ENTRY PERSON-REC IS RECORD PERSON-REC
```

2. Every item in the logical record definition is included in the entry definition (i.e., there is no ITEMS ARE: clause in a default ADF)

3. Entry access paths are determined by the prime retrieval path (given by the RETRIEVAL VIA clause in a 01-level definition) of the logical record comprising the entry. Entries are created as Primary and Thru entries accordingly. Thus a partial DDF of

```
01   REC-A
     ; TYPE IS "A" IN TYPE-FIELD
     ; RETRIEVAL VIA SCAN.
     ---
98   A-B MASTER
01   REC-B
     ; TYPE IS "B" IN TYPE-FIELD
     ; RETRIEVAL VIA A-B.
     ---
98   A-B DETAIL
```

would result in entry definitions

```
ENTRY REC-A IS RECORD REC-A
ENTRY REC-B IS RECORD REC-B
      THRU REC-A
```

In addition, logical records in an ISP or Inverted file defined with a RETRIEVAL VIA SEARCH clause, and IDS logical records defined with a RETRIEVAL VIA CALC CHAIN clause, will result in entries defined as VIA SEARCH. Thus, ISP, Inverted, and IDS calculating values do not have to be supplied when these entries are extracted from the data file.

20-5. The simplicity of the Default ADF approach is obvious. A data item exists in one and only one entry. The retrieval path to that entry is uniquely constructed from the prime retrieval path for the single record comprising the entry.

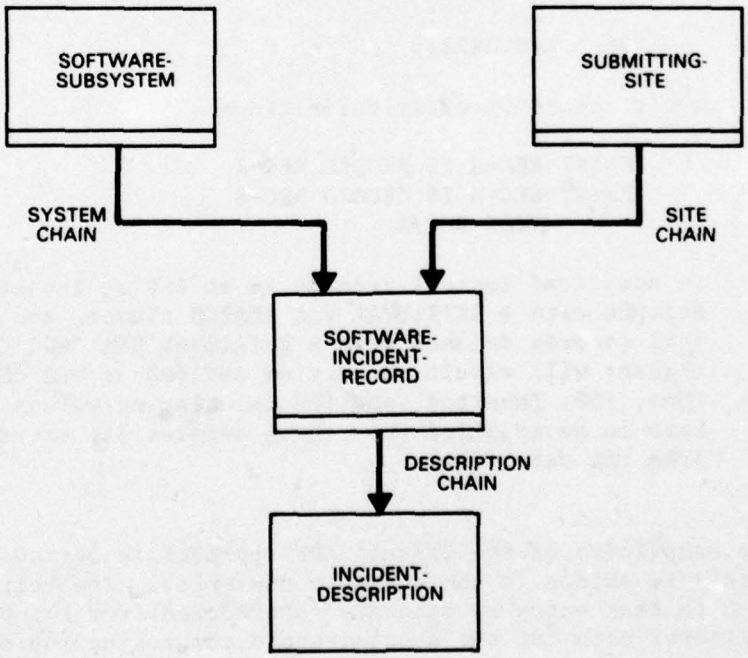
Thus, a data item called for in a QUERY procedure can be accessed in one and only one way using a default ADF. All conflicting retrieval paths to the logical record containing the data item are ignored.

20-6. The Default ADF approach is well-tailored to data base structures for which the following conditions hold true:

1. The structure is hierarchical (as opposed to network) in nature.

2. Only those records which are at the top of a hierarchy (i.e., those that are not details of any set) are defined in the DDF with a RETRIEVAL VIA SCAN, RETRIEVAL VIA SEARCH, or RETRIEVAL VIA CALC clause. All others are defined with RETRIEVAL VIA chain-name.

If either or both of these conditions does not hold true, a default ADF may not necessarily be the best ADF to use for a particular application. This is especially critical when dealing with IDS structures. For example, note the following IDS structure used to track software incidents.



In this file, data processing incidents are logged by major computer subsystems, by computer installation that encountered the problem, and by incident identification. The intention is that incident records are to be accessed by one of the following methods:

- Supply an incident identification value and CALC directly to that incident record;
- Supply a software subsystem name (e.g. FORTRAN) and walk the SYSTEM chain, retrieving all or only certain related INCIDENT record instances;
- Supply the designation for a particular computer installation, and walk the SITE chain, retrieving some or all of the incident records generated by that site.

The following entries will be generated by the default ADF generation program (records SOFTWARE-SUBSYSTEM, SUBMITTING-SITE, and SOFTWARE-INCIDENT-RECORD are CALC records):

```
ENTRY SOFTWARE-SUBSYSTEM IS RECORD
      SOFTWARE-SUBSYSTEM VIA SEARCH
ENTRY SUBMITTING-SITE IS RECORD
      SUBMITTING-SITE VIA SEARCH
ENTRY SOFTWARE-INCIDENT-RECORD IS RECORD
      SOFTWARE-INCIDENT-RECORD VIA SEARCH
ENTRY INCIDENT-DESCRIPTION IS RECORD
      INCIDENT-DESCRIPTION THRU
          SOFTWARE-INCIDENT-RECORD
```

This default ADF works fine if we are interested in either retrieving a specific incident record, or retrieving some or all incident records without regard to the master SOFTWARE-SUBSYSTEM or SUBMITTING-SITE records, since VIA SEARCH represents the access path to the SOFTWARE-INCIDENT-RECORD records.

If, however, instances of both the SOFTWARE-SUBSYSTEM and SOFTWARE-INCIDENT-RECORD records were desired, there would be no way to guarantee that these instances would be related to one another. The same would be true if it were desired to process instances of both the SUBMITTING-SITE and SOFTWARE-INCIDENT-RECORD records.

The best way that the Data Base Administrator could insure that all three methods of retrieval of the SOFTWARE-INCIDENT-RECORD record would still be available would be to

supply multiple ADFs to access the data base. Each user could then choose from among the available ADFs the one best adapted to his particular application.

For example, the DBA could provide:

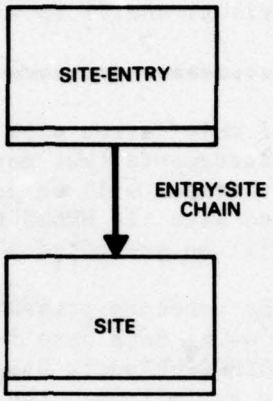
1. A default ADF, suitable for retrieving SOFTWARE-INCIDENT-RECORD records without regard to master SOFTWARE-SUBSYSTEM or SUBMITTING-SITE records.

2. An ADF suitable for retrieving each SOFTWARE-INCIDENT-RECORD record associated with a given SOFTWARE-SUBSYSTEM record, and then retrieving the master SUBMITTING-SITE record associated with the SOFTWARE-INCIDENT-RECORD record. The following ADF would accomplish this purpose:

```
ENTRY  SOFTWARE-SUBSYSTEM IS RECORD
       SOFTWARE-SUBSYSTEM VIA SEARCH
ENTRY  INCIDENT-AND-SITE IS RECORD
       SOFTWARE-INCIDENT-RECORD THRU
       SOFTWARE-SUBSYSTEM VIA SYSTEM
       AND SUBMITTING-SITE VIA SITE
```

3. An ADF suitable for retrieving each SOFTWARE-INCIDENT-RECORD record associated with a given SUBMITTING-SITE record, and then retrieving the master SOFTWARE-SUBSYSTEM record associated with the SOFTWARE-INCIDENT-RECORD record.

20-7. Default ADFs are, in general, not very useful for accessing IDS data bases, even those that are hierarchical in structure. Recall that records defined with a RETRIEVAL VIA CALC clause in the DDF result in entries defined as VIA SEARCH. While this definition may have some useful applications, it can also work as a major drawback leading to gross inefficiency. To see why this is true, note the following structure.



There is exactly one SITE-ENTRY record in the data base and it is accessed by CALCing in on the value "SITES". The sole purpose of this record is to allow the user to retrieve all of the SITE records in logical order simply by walking a chain (the ENTRY-SITE chain). The following ADF entry provides a means for accomplishing this using WWDMS:

```
ENTRY SITE IS RECORD SITE-ENTRY VIA CALC
    AND RECORD SITE VIA ENTRY-SITE
```

Note, however, what would happen if we were to attempt to retrieve all of the SITE records using the following default ADF entries.

```
ENTRY SITE-ENTRY IS RECORD
    SITE-ENTRY VIA SEARCH
ENTRY SITE IS RECORD SITE VIA SEARCH
```

Note that the ENTRY-SITE chain is ignored in the default ADF, since VIA SEARCH, not VIA ENTRY-SITE, represents the access path to the SITE record. In order to retrieve all of the SITE records, WWDMS must search every page of the data file for SITE records. Not only is this extremely inefficient, but the SITE records would be retrieved according to their physical order in the data file, which may not correspond at all to their logical order.

20-8. TRUE or FALSE:

Standard (non-default) ADFs should not be used in a QUERY procedure

FALSE. Standard (non-default) ADFs can be used if no record retrieval conflicts will result.

●●●

```
000010  MD   COMPANY-FILE;  FILE IS SEQUENTIAL.
000020  01   COMPANY-REC
000030           ;TYPE IS "101" IN COMPANY-TYPE
000040           ;RETRIEVAL VIA SCAN.
000050    02   COMPANY-SEQUENCE-FIELD.
000060      03   COMP-SITE-NO                          PIC 999.
000070      03   CODE-101                              PIC X.
000080      03   CODE-102                              PIC X.
000090      03   CODE-103                              PIC X.
000100    02   COMPANY-TYPE                          PIC XXX.
000110    02   COMPANY-NAME                          PIC X(25).
000120    02   COMPANY-INITIALS                      PIC XXX.
000130    02   PRESIDENT                             PIC X(20).
000140    02   HEADQUARTERS-ADDRESS.
000150      03   STREET-ADDRESS.
000160         04   NUMBER                               PIC 9(5).
000170         04   STREET-NAME                          PIC X(20).
000180      03   CITY-STATE-ADDRESS.
000190         04   CITY-STATE.
000200           05   CITY                                 PIC X(15).
000210           05   STATE                                PIC XX.
000220         04   ZIP-CODE                             PIC 9(5).
000230  98   SYSTEM DETAIL ; SCAN ON COMPANY-SEQUENCE-FIELD.
000240  98   COMPANY-SITE MASTER ; ORDER IS SORTED.
000250  01   SITE-REC
000260           ;TYPE IS "201" IN SITE-TYPE
000270           ;RETRIEVAL VIAL SCAN.
000280    02   SITE-SEQUENCE-FIELD.
000290      03   SITE-SITE-NO                          PIC 999.
000300      03   CODE-201                              PIC X.
000310      03   CODE-202                              PIC X.
000320      03   CODE-203                              PIC X.
000330    02   SITE-TYPE                             PIC XXX.
000340    02   SITE-DESIGNATION                      PIC X(25).
000350    02   SITE-NUMBER                           PIC 999.
000360    02   SITE-MANAGER                          PIC X(20).
000370    02   SITE-LOCATION.
000380      03   SITE-STREET                           PIC X(25).
000390      03   SITE-CITY-STATE                       PIC X(22).
000400  98   SYSTEM DETAIL; SCAN ON SITE-SEQUENCE-FIELD.
000410  98   COMPANY-SITE DETAIL
000420           ;ASCENDING KEY IS SITE-SEQUENCE-FIELD.
000430  98   SITE-PERSONNEL MASTER; ORDER IS SORTED.
000460  01   SUPERVISORY-PERSONNEL-REC
000470           ;TYPE IS "301" IN SUPER-TYPE
000480           ;RETRIEVAL VIA SITE-PERSONNEL.
000490    02   SUPERVISOR-SEQUENCE-FIELD.
000500      03   SUPER-SITE-NO                         PIC 999.
000510      03   CODE-301                              PIC X.
000520      03   CODE-302                              PIC X.
```

Illustration 20.1 Data Definition Source File

```
000530      03   CODE-303                                  PIC X.
000540    02   SUPER-TYPE                                PIC XXX.
000550    02   SUPERVISOR-NAME                           PIC X(30).
000560    02   SUPER-SOCIAL-SECURITY-NO                  PIC X(11).
000570    02   SUPERVISOR-HOME-ADDRESS.
000580      03   SUPER-STREET                             PIC X(25).
000590      03   SUPER-CITY-STATE                         PIC X(22).
000600    02   SUPERVISOR-HOME-PHONE.
000610      03   SUPER-AREA-CODE                          PIC 999.
000620      03   SUPER-PHONE-NUMBER                       PIC 9(7).
000630    02   DATE-SUPER-ENTERED-COMPANY                PIC 9(6).
000640    02   DATE-CURRENT-POSITION-SUPER               PIC 9(6).
000650  98  SITE-PERSONNEL DETAIL
000660           ;ASCENDING KEY IS SUPERVISOR-SEQUENCE-FIELD.
000670  98  BOSS-WORKER MASTER; ORDER IS SORTED.
000680  01  CLERICAL-PERSONNEL-REC
000690           ;TYPE IS "401" IN CLER-TYPE
000700           ;RETRIEVAL VIA BOSS-WORKER.
000710    02   CLERICAL-SEQUENCE-FIELD.
000720      03   CLER-SITE-NO                             PIC 999.
000730      03   CODE-401A                                PIC X.
000740      03   CODE-401B                                PIC X
000750      03   CODE-401C                                PIC X
000760    02   CLER-TYPE                                 PIC XXX.
000770    02   CLERICAL-NAME                             PIC X(30).
000780    02   CLERICAL-SOCIAL-SECURITY-NO               PIC X(11).
000790    02   CLERICAL-HOME-ADDRESS.
000800      03   CLERICAL-STREET                          PIC X(25).
000810      03   CLERICAL-CITY-STATE                      PIC X(22).
000820    02   CLERICAL-HOME-PHONE.
000830      03   CLERICAL-AREA-CODE                       PIC 999.
000840      03   CLERICAL-PHONE-NUMBER                    PIC 9(7).
000850    02   DATE-CLERICAL-ENTERED-COMPANY             PIC 9(6).
000860    02   CLERICAL-WAGE-INFO.
000870      03   CURRENT-HOURLY-WAGE                      PIC 99V99.
000880      03   PREVIOUS-HOURLY-WAGE                     PIC 99V99.
000890    02   CLERICAL-SKILL-INFO.
000900      03   SHORTHAND-SPEED-WPM                      PIC 999.
000910      03   TYPING-SPEED-WPM                         PIC 999.
000920  98  BOSS-WORKER DETAIL
000930           ;ASCENDING KEY IS CLERICAL-SEQUENCE-FIELD.
000940  01  TECHNICAL-PERSONNEL-REC
000950           ;TYPE IS "402" IN TECH-TYPE
000960           ;RETRIEVAL VIA BOSS-WORKER.
000970    02   TECHNICAL-SEQUENCE-FIELD.
000980      03   TECH-SITE-NO                             PIC 999.
000990      03   CODE-402A                                PIC X.
001000      03   CODE-402B                                PIC X.
001010      03   CODE-402C                                PIC X.
001020    02   TECH-TYPE                                 PIC XXX.
```

Illustration 20.1 (Continued)

```
001030     02   TECHNICAL-NAME                          PIC X(30).
001040     02   TECHNICAL-SOCIAL-SECURITY-NO            PIC X(11).
001050     02   TECHNICAL-HOME-ADDRESS.
001060       03   TECHNICAL-STREET                       PIC X(25).
001070       03   TECHNICAL-CITY-STATE                   PIC X(22).
001080     02   TECHNICAL-HOME-PHONE.
001090       03   TECHNICAL-AREA-CODE                    PIC 999.
001100       03   TECHNICAL-PHONE-NUMBER                 PIC 9(7).
001110     02   DATE-TECHNICAL-ENTERED-COMPANY          PIC 9(6).
001120     02   DATE-CURRENT-ASSIGNMENT                 PIC 9(6).
001130     02   TECHNICAL-WAGE-INFO.
001140       03   CURRENT-MONTHLY-WAGE                   PIC 9(4).
001150       03   PREVIOUS-MONTHLY-WAGE                  PIC 9(4).
001160     02   TECHNICAL-SKILL-INFO.
001170       03   SKILL-CODE                             PIC 99.
001180       03   NUMBER-OF-YEARS-OF-SKILL               PIC 99.
001190       03   DATE-COMPUTED                          PIC 9(6).
001200       03   SKILL-LEVEL-CODE                       PIC X.
001210   98  BOSS-WORKER DETAIL
001220           ;ASCENDING KEY IS TECHNICAL-SEQUENCE-FIELD.
```

Illustration 20.1 (Continued)

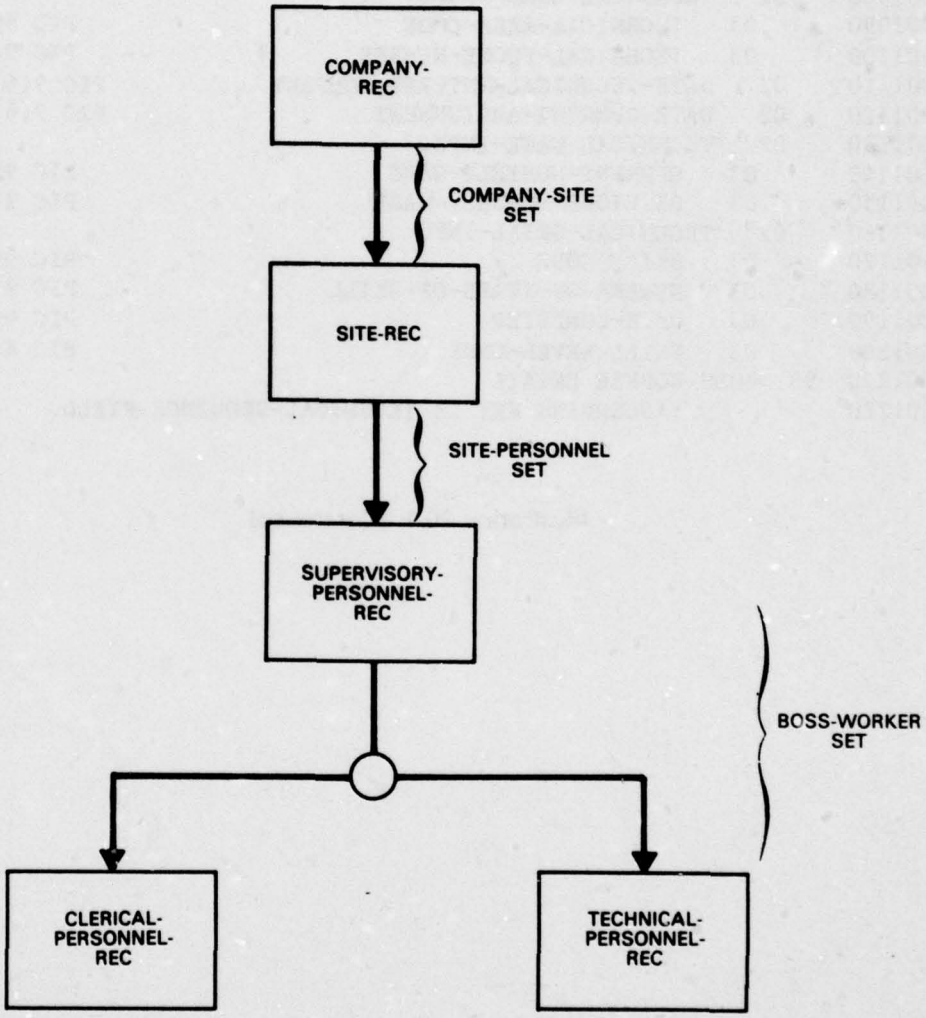


Illustration 20.2 Data Base Structure Graph

```
010     DATA-BASE IS COMPANY-STRUCTURE IN PRC606/COMP-DIR
020         ENTRY COMPANY IS RECORD COMPANY-REC
030         ENTRY SITE IS RECORD SITE-REC
040             THRU COMPANY VIA COMPANY-SITE
050         ENTRY SUPERVISOR IS RECORD SUPERVISORY-PERSONNEL-REC
060             THRU SITE
070         ENTRY CLERICAL-EMPLOYEE
080             IS RECORD CLERICAL-PERSONNEL-REC
090             THRU SUPERVISOR
100         ENTRY TECHNICAL-EMPLOYEE
110             IS RECORD TECHNICAL-PERSONNEL-REC
120             THRU SUPERVISOR
```

Illustration 20.3. Application Definition Source File

The optional conditional expression functions similarly to a RETRIEVE statement conditional expression. It "selects" instances of retrieved items for processing from the total set of retrieved instances.

20-12. Data items used in a QUERY statement conditional clause must either:

1. Appear explicitly in the procedure's LINE, SORT, LET, and PRINT statements, or

2. Be contained in those records which contain items which appear explicitly in the procedure's LINE, SORT, LET, and PRINT statements, or

3. Be contained in records which lie in the retrieval path of those records which contain items which appear explicitly in the procedure's LINE, SORT, LET, and PRINT statements.

Some examples of valid and invalid QUERY statement conditionals are shown below. The data base being queried is that shown in Illustration(s) 20.1, 20.2, and 20.3.

Example 1

```
010  QUERY PRC606/COMPADFO
020  WHERE COMPANY-INITIALS = "ABC"
030  PRINT ON PRINTER COMPANY-INITIALS,
040  COMPANY-NAME
```

This is a valid QUERY statement conditional, since data item COMPANY-INITIALS appears explicitly in the procedure body.

Example 2

```
010  QUERY PRC606/COMPADFO
020  WHERE SITE-NUMBER = 10 OR 106
030        OR 255 OR 999
040  PRINT ON PRINTER SITE-DESIGNATION.
```

This is a valid QUERY statement conditional, since data item SITE-NUMBER appears in the same logical record as the item SITE-DESIGNATION, which is explicitly mentioned in a PRINT statement.

Example 3

```
010  QUERY PRC606/COMPADFO
020  WHERE COMPANY-INITIALS = "ABC"
030  PRINT ON PRINTER SITE-DESIGNATION
```

This is a valid QUERY statement conditional. Although the conditional identifier COMPANY-INITIALS is not contained in the same logical record as SITE-DESIGNATION, logical record COMPANY-REC must be retrieved before logical record SITE-REC. Thus, identifier COMPANY-INITIALS lies in a logical record that is in the retrieval path of the logical record containing SITE-DESIGNATION, and is legal in the QUERY conditional.

Example 4

```
010  QUERY PRC606/COMPADFO
020  WHERE SITE-NUMBER = 95
030  PRINT ON PRINTER COMPANY-NAME
```

This is an invalid QUERY statement conditional. SITE-NUMBER lies in logical record SITE-REC, which is not in the retrieval path of logical record COMPANY-REC, which contains the explicitly named data item COMPANY-NAME.

20-13. This rule for QUERY conditionals applies to all data bases, regardless of the ADF used to access the data base. For example, assume that the source Application Definition File PRC606/COMPADFO contained the following entry definition only:

```
DATA-BASE IS COMPANY-STRUCTURE IN PRC606/COMP-DIR
  ENTRY ALL-RECS IS RECORD COMPANY-REC
    AND SITE-REC VIA COMPANY-SITE
      AND SUPERVISORY-PERSONNEL-REC
```

The following WWDMS procedure is valid.

```
010  INVOKE PRC606/COMPADFO
020  RETRIEVE ALL-RECS FROM COMPANY-STRUCTURE
030    WHERE SUPERVISOR-NAME CONTAINS "SMITH"
040  PRINT ON PRINTER SITE-DESIGNATION
050  END
```

Consider the following QUERY procedure.

```
010  QUERY PRC606/COMPADFO
020    WHERE SUPERVISOR-NAME CONTAINS "SMITH"
030  PRINT ON PRINTER SITE-DESIGNATION
040  END
```

At first glance, it would appear that the above procedure is valid, since only one entry exists in the ADF being invoked, and the QUERY procedure must use that three-level hierarchical entry as it retrieves data from the data file. Nevertheless, an attempt to run this procedure would produce unexpected results.

20-14. PRINT statement.

The QUERY procedure PRINT statement, like its WWDMS counterpart, causes the displaying of one or more lines of a report. However, several syntactical and functional differences do exist between the QUERY and WWDMS PRINT statements.

There are three forms of the QUERY Procedure PRINT statement:

1. PRINT ON...

2. PRINT label , label...

3. PRINT report-name

You are already familiar with the WWDMS versions of the PRINT ON... and PRINT label statements. The PRINT report-name statement is not valid in a WWDMS procedure; it is unique to QUERY.

In the frames that follow, we will discuss the syntax and function of the most basic form of the PRINT statement, the PRINT ON... statement. The PRINT label and PRINT report-name statements are used in the generation of more sophisticated reports and will be discussed in the context of the REPORT statement.

20-15. The QUERY procedure PRINT ON. . . statement has the following format:

```
PRINT
[ PRINTER  { DUP       }  n                                                    ]
           { DUPLICATE }
[ REMOTE   id [ { DUP       } n  [,id { DUP       } n]. . .]                  ]
           [ { DUPLICATE }          { DUPLICATE }                             ]
[ ON       { FILE   catalog/file-string [, catalog/file-string]              } ]
           [ FOR { TTY                                                       } ]
                 { CRT                                                       }
                 { PRINTER [IN BCD]                                          }
                 { terminal-name                                             }
[ DAC                                                                        ]
[SCC IS literal]
report-elem-descr [,report-elem-descr] ...
```

This is nearly exactly the same as the format of the WWDMS PRINT ON... statement, which was discussed in Lessons 4 and 11. The one syntactical difference between the QUERY PRINT ON... statement and its WWDMS counterpart is that it is not possible to specify an entry name in the QUERY PRINT ON... statement. Recall from Frame 20-2 that entry names are not valid in a QUERY procedure.

20-16. Although the QUERY and WWDMS PRINT ON... statements are used for basically the same purpose, there is a slight functional difference between the two. In order to explain this difference, it is necessary to introduce at this time the concept of record availability.

Referencing Illustrations 20.1, 20.2, and 20.3, consider the following QUERY procedure:

```
010  QUERY PRC606/COMPADFO
020  PRINT ON PRINTER SITE-NUMBER, SUPERVISOR-NAME,
030    CLERICAL-NAME, TECHNICAL-NAME
```

There are two retrieval paths which must be traversed in order to obtain the items referenced by the PRINT statement:

1. The path from COMPANY-REC to SITE-REC to SUPERVISORY-PERSONNEL-REC to CLERICAL-PERSONNEL-REC.

2. The path from COMPANY-REC to SITE-REC to SUPERVISORY-PERSONNEL-REC to TECHNICAL-PERSONNEL-REC.

A QUERY PRINT ON. . . statement is executed each time the lowest level record on any retrieval path has been retrieved. A record is available at the time of the PRINT if it lies on the retrieval path of the lowest level record retrieved. Data items in records which are not available at the time of the PRINT are displayed as absent (blanks for alphanumeric items, zeros for numeric items).

In terms of our procedure, this means:

1. The PRINT is executed each time a CLERICAL-PERSONNEL-REC record or TECHNICAL-PERSONNEL-REC record is retrieved.

2. When the PRINT is executed following the retrieval of a CLERICAL-PERSONNEL-REC record, TECHNICAL-PERSONNEL-REC is not available and data item TECHNICAL-NAME is displayed as blank.

3. Similarly, when the PRINT is executed after a TECHNICAL-PERSONNEL-REC record has been retrieved, CLERICAL-PERSONNEL-REC is not available and data item CLERICAL-NAME is displayed as blank.

Thus, the output from the procedure would appear as follows:

SITE-NUMBER	SUPERVISOR-NAME	CLERICAL-NAME	TECHNICAL-NAME
001	JOHN A SMITH	J B CHRISTENSEN	
001	JOHN A SMITH		R Q FAHNLANDER
001	JOHN A SMITH		F X EARL
001	WILLIAM B WILLIAMS	M N JACKSON	

```
001          WILLIAM B WILLIAMS                      B R ATLAND
001          WILLIAM B WILLIAMS                      J R COOK
001          WILLIAM B WILLIAMS                      T A ZELAND
002          CLARK S KENT          D J BYRON
002          CLARK S KENT                            T J KELLY
002          CLARK S KENT                            T R ERICKSON
```

20-17. Consider the following procedure:

```
010  QUERY PRC606/COMPADFO
020  PRINT ON PRINTER SITE-DESIGNATION,
030    SUPERVISOR-NAME, CLERICAL-NAME,
040    TYPING-SPEED-WPM, TECHNICAL-NAME
```

Which one of the five data items referenced in the PRINT statement will not be available at the time at which all of the other four items are available?

●●●

●●●

20-23. REPORT statement.

The QUERY procedure REPORT statement serves the same funcion as its WWDMS counterpart; that is, it is used to specify the format and layout of an output report. The format of the REPORT statement is as follows:

```
REPORT report-name

[ON { PRINTER {DUPLICATE} n                                                    }
   {          {DUP      }                                                     }
   {                                                                          }
   { REMOTE id{DUPLICATE} n [,id{DUPLICATE} n]...                             }
   {         {DUP       }      {DUP      }                                   }
   {                                                                          }
   { FILE catalog/file-string [,catalog/file-string]...                       }
   {                                                                          }
   { [FOR {TTY                    }]                                          }
   {      {CRT                    }                                           }
   {      {PRINTER [IN BCD]       }                                           }
   {      {terminal-name          }                                           }
   {                                                                          }
   { DAC                                                                      }]

[SCC IS literal]

[{COLUMN} SPACING IS nn]
 {COL   }

[PAGE LENGTH nnn LINES]

[PAGE WIDTH nnn CHARACTERS]

[NUMBER OF PAGES nnnnnn]

[{STARTING PAGE NUMBER} [TOTAL PAGES [nnnnnn]]]
 {NO PAGE NUMBER       }              [literal]

[COVER PAGE IS label [,label]...]

[PAGE HEADING IS label [, label]...]

[PAGE FOOTING is label [, label]...]

[CONTROL HEADING FOR label-1 IS label-2 [,label-n] . . .
```

```
                                  ||identifier-1 [,identifier-2]...CHANGE||
        BREAK ON                  ||                                     ||  ]
                                  ||FIRST                                ||

[CONTROL FOOTING FOR label-3 IS label-4 [,label-n]...

                                  ||identifier-3[,identifier-4]...CHANGE||
        BREAK ON                  ||                                     ||  ]
                                  ||LAST                                 ||
```

The QUERY REPORT statement functions in exactly the same manner as its WWDMS counterpart, with the following exception: if the ON clause is not specified, the report is displayed on the user's terminal.

20-24. A QUERY report body, like a WWDMS report body, is made up of one or more report-packets. Recall from Lesson 12 that a report-packet, by definition, consists of a labeled LINE or SPACE statement and any unlabeled LINE or SPACE statements that immediately follow it.

Those report-packets which are not output implicitly (i.e., *those that are* not part of the cover page, page headings or footings, or control headings or footings) are called detail-line sets and must be referenced by PRINT label or PRINT report-name statements.

20-25. PRINT label statement.

The PRINT label statement is used to specify the output of a given detail-line set. The QUERY PRINT label statement has the same format as its WWDMS counterpart, to which you were introduced in Lesson 12. The format is:

```
PRINT label  ,label  ...
```

20-26. PRINT report-name statement.

The PRINT report-name statement specifies that all of the detail-line sets defined for the named report-body are to be displayed at the appropriate time. The format of this type of PRINT statement is as follows:

```
PRINT report-name
```

The PRINT report-name has the same effect as a PRINT label statement which references all of the detail-line sets in the report body. Thus, the PRINT statements in the following two procedures are equivalent.

```
010  QUERY
020  REPORT R1
030       COVER PAGE IS CVR-PG
040       PAGE HEADING IS PG-HDG.
050       PAGE FOOTING IS PG-FTG.
060  CVR-PG.
070  PG-HDG.
080  PG-FTG.
090  DTL-1.
100  DTL-2.
110  DTL-3.
120  PRINT DTL-1, DTL-2, DTL-3.
130  END
```

```
010  QUERY
020  REPORT R1
030       COVER PAGE IS CVR-PG.
040       PAGE HEADING IS PG-HDG.
050       PAGE FOOTING IS PG-FTG.
060  CVR-PG.
070  PG-HDG.
080  PG-FTG.
090  DTL-1.
100  DTL-2.
110  DTL-3.
120  PRINT R1
130  END
```

The PRINT report-name statement is unique to QUERY and is not valid in full WWDMS procedures.

20-27. There is a fundamental difference in function between the WWDMS PRINT label statement, on the one hand, and the QUERY PRINT label and PRINT report-name statements, on the other.

The WWDMS PRINT label statement is executed each time control passes through it. With each such execution, all of the report lines which are contained in the referenced detail-line set are output.

The QUERY PRINT label and PRINT report-name statements, on the other hand, do not directly cause the output of these report lines, but merely serve to specify which lines are to be output at the appropriate time. The time at which a given

line of a detail-line set is displayed is dependent upon the data items included in that line. Each line of a detail-line set is treated separately and may or may not be displayed at the same time as the other lines contained in the same detail-line set.

The rules regarding the output for report lines will be explained in the following frames on the LINE and SPACE statements.

20-28. LINE statement.

The LINE statement has the same function in a QUERY procedure as it does in a WWDMS procedure: to specify the content and format of a report line.

Recall that the WWDMS procedure LINE statement has the following format:

```
LINE  { entry-name          } [ ,report-elem-descr]...
      { report-elem-descr   }
```

In contrast, the QUERY procedure LINE statement has the following format:

```
LINE report-elem-descr [,report-elem-descr]...
```

Note that it is not possible to specify an entry name in a QUERY procedure LINE statement.

20-29. LINE statements which contain data items are displayed in the same manner as PRINT ON...statements; that is, the line is displayed each time the lowest level record on any retrieval path has been retrieved. Thus, the following two QUERY procedures will result in the display of the same detail lines:

```
010  QUERY PRC606/COMPADFO
020  PRINT ON PRINTER COMPANY-INITIALS,
030       COMPANY-NAME, SITE-DESIGNATION
040  END
```

```
010  QUERY PRC606/COMPADFO
020  REPORT R1 ON PRINTER
030  DTL-1.  LINE COMPANY-INITIALS,
040       COMPANY-NAME, SITE-DESIGNATION.
050  PRINT R1
060  END
```

20-30. If there are two or more LINE statements which contain data items, then each is output independent of the other(s) depending on the items that line itself contains. As an example, consider the following procedure:

```
010  QUERY PRC606/COMPADFO
020  REPORT R1 ON PRINTER
030  L1.  LINE COMPANY-NAME, PRESIDENT.
040       LINE SITE-DESIGNATION, SITE-MANAGER.
050  L2.  LINE SUPERVISOR-NAME, CLERICAL-NAME,
060       TECHNICAL-NAME.
070  PRINT R1
080  END
```

The LINE statement at line 030 is displayed whenever a new COMPANY-REC record is retrieved.

The LINE statement at line 040 is displayed whenever a new SITE-REC record is retrieved.

The LINE statement at line 050 is displayed whenever a new CLERICAL-PERSONNEL-REC or TECHNICAL-PERSONNEL-REC record is retrieved.

20-31. Several additional notes on the procedure in the preceding frame:

1. Note that the LINE statement at line 030 is labeled. Any LINE statement which immediately follows a REPORT statement must be labeled.

2. The fact that the LINE statement at line 030 is labeled and the one at line 040 is not labeled is irrelevant. All LINE statements which contain data items, except one that is the first LINE statement after a REPORT statement, may or may not be labeled.

20-32. LINE statements which do not contain data items are displayed according to the following rules:

1. If the statement is labeled, then it is executed prior to the next LINE statement in the same detail-line set that contains data items.

2. If the statement is unlabeled, then it is executed after the last preceding LINE statement in the same detail-line set that contains data items.

```
010  QUERY PRC606/COMPADFO
020  REPORT R1
030  DTL-1.  LINE "THIS IS LINE A".
040          LINE COMPANY-INITIALS,
050               COMPANY-NAME.
060  DTL-2.  LINE SITE-DESIGNATION.
070          LINE "THIS IS LINE B".
080  PRINT R1
090  END
```

In the above procedure, the line reading "THIS IS LINE A" will be printed before each display of the line containing the items COMPANY-INITIALS and COMPANY-NAME. The line reading "THIS IS LINE B" will be printed after each display of the line containing the item SITE-DESIGNATION.

20-33. SPACE statement. The QUERY procedure SPACE statement has the same function as its WWDMS counterpart: to control slewing in an output report. The format is also the same and is repeated below:

$$\text{SPACE} \quad \left\{ \begin{matrix} nn \\ \underline{TOP} \end{matrix} \right\}$$

The rules for the display of SPACEs are the same as the rules for the display of LINE statements that do not contain data items.

20-34. The QUERY Procedure Language, like the WWDMS Procedure Language, contains a SORT statement which allows the user to order data prior to printing. As was the case with the PRINT statement, there are several syntactical and functional differences between the QUERY SORT statement and the WWDMS SORT statement. We will examine the syntactical differences first.

20-35. If a SORT statement is to be present in the procedure, it must be placed after any report bodies but before any PRINT statements that are to reference the sorted data. The format of the SORT statement is as follows:

$$\underline{\text{SORT}} \quad \left[\left\{ \begin{matrix} \text{report-name} \\ \text{identifier} \end{matrix} \right\} \right] \quad [,\text{identifier}]\ldots]$$

$$\underline{\text{ON}}\ \text{sort-key-1}\ \ \text{IN}\left[\left\{ \begin{matrix} \underline{\text{ASC}} \\ \underline{\text{ASCENDING}} \\ \underline{\text{DSC}} \\ \underline{\text{DESCENDING}} \end{matrix} \right\} \quad \left\{ \begin{matrix} \text{ORDER} \\ \text{SEQUENCE} \end{matrix} \right\} \right]$$

```
[,sort-key-n  [IN { ASC        }      { ORDER    } ] ...]
                  { ASCENDING  }      { SEQUENCE }
                  { DSC        }
                  { DESCENDING }
```

There are three basic syntactical differences between the QUERY SORT statement and the WWDMS SORT statement:

1. Only one SORT statement is allowed within a QUERY procedure.

2. The SORT entry-name statement is not valid in a QUERY procedure.

3. The SORT report-name statement is valid in QUERY, but not in WWDMS.

20-36. Like the WWDMS SORT, the QUERY SORT consists of two subphases. The first or "hold" subphase writes the data instances to a temporary file in sorted order. The second or "deferred" subphase delivers the sorted data instances for subsequent processing.

What exactly do these "data instances" look like? That depends on the type of SORT statement used. We will differentiate among the three types of SORT statement a little later, but for now you should simply be aware that, regardless of the type of SORT statement used, each data instance will consist of:

1. All of the sort-keys specified in the SORT statement.

2. Some, but ordinarily all, of the identifiers specified in the procedure's LINE and PRINT statements.

20-37. During the "hold" subphase, QUERY retrieves the lowest level record along each retrieval path, as if it were developing a standard report. Each time QUERY has retrieved enough information to print a new report line, a data instance is constructed and written to the temporary file.

All sort-keys specified in the SORT statement must be available at each time at which a data instance is constructed. Those identifiers which appear in the procedure's LINE and PRINT statements, but which are not sort-keys may or may not be available; if they are not available then QUERY will mark them as such in the sort data instance. Examples of the construction of data instances are shown below.

Example 1

```
010  QUERY PRC606/COMPADFO
020  SORT ON COMPANY-INITIALS
030  PRINT ON PRINTER COMPANY-NAME,
040       SITE-MANAGER, SUPERVISOR-NAME
050  END
```

A data instance will be constructed each time that QUERY has retrieved enough information to construct a print line; that is, each time a new SUPERVISORY-PERSONNEL-REC record is retrieved. Since the sort-key COMPANY-INITIALS is always available, the sort-key rule is not violated.

Example 2

```
010  QUERY PRC606/COMPADFO
020  REPORT R1 ON PRINTER
030  L1.  LINE COMPANY-NAME,
040       PRESIDENT.
050  L2.  LINE SITE-DESIGNATION,
060       SITE-MANAGER.
070  SORT R1 ON COMPANY-INITIALS
080  PRINT R1
```

A data instance will be constructed under either of the following circumstances:

1. The LINE statement at line 030 is satisfied; that is, each time a new COMPANY-REC record is retrieved.

2. The LINE statement at line 050 is satisfied; that is, each time a new SITE-REC record is retrieved.

Let's step through our procedure to see exactly what happens after each retrieval. Assume that our data base consists of the following records:

RECORD TYPE	COMPANY-INITIALS	COMPANY-NAME	PRESIDENT	SITE-DESIGNATION	SITE-MANAGER
COMPANY-REC	ABC	ABC COMPANY	G. WASHINGTON	----------	----------
SITE-REC	----------	----------	----------	WASHINGTON	JOHN A SMITH
SITE-REC	----------	----------	----------	PENTAGON	W WILLIAMS
COMPANY-REC	DEF	DEF COMPANY	J ADAMS	----------	----------
SITE-REC	----------	----------	----------	RESTON	CLARK S KENT

Retrieval of the first COMPANY-REC record satisfies the first LINE statement, so a data instance is constructed. Since items SITE-DESIGNATION and SITE-MANAGER

are not yet available, they are marked as such in the data instance. Thus, the data instance would appear as follows:

COMPANY-INITIALS	COMPANY-NAME	PRESIDENT	SITE-DESIGNATION	SITE-MANAGER
ABC	ABC COMPANY	G. WASHINGTON	N/A	N/A

The next record, a SITE-REC record, is then retrieved. This satisfies the second LINE statement, so another data instance is constructed. Note that all items are now available. The data instance would appear as follows:

COMPANY-INITIALS	COMPANY-NAME	PRESIDENT	SITE-DESIGNATION	SITE-MANAGER
ABC	ABC COMPANY	G. WASHINGTON	WASHINGTON	JOHN A SMITH

Retrieval of the next record, also a SITE-REC record, causes the following data instance to be constructed.

COMPANY-INITIALS	COMPANY-NAME	PRESIDENT	SITE-DESIGNATION	SITE-MANAGER
ABC	ABC COMPANY	G. WASHINGTON	PENTAGON	W WILLIAMS

Retrieval of the next record, a COMPANY-REC record, causes the first LINE statement to be satisfied. The items SITE-DESIGNATION and SITE-MANAGER are again not available. The constructed data instance would appear as follows:

COMPANY-INITIALS	COMPANY-NAME	PRESIDENT	SITE-DESIGNATION	SITE-MANAGER
DEF	DEF COMPANY	J ADAMS	N/A	N/A

Finally, retrieval of the last record, a SITE-REC record, causes the following data instance to be constructed.

COMPANY-INITIALS	COMPANY-NAME	PRESIDENT	SITE-DESIGNATION	SITE-MANAGER
DEF	DEF COMPANY	J ADAMS	RESTON	CLARK S KENT

Note again that since the sort-key, COMPANY-INITIALS, is always available the sort-key rule is not violated.

Example 3

```
010  QUERY PRC606/COMPADFO
020  REPORT R1 ON PRINTER
030  L1.  LINE COMPANY-NAME,
040       PRESIDENT.
050  L2.  LINE SITE-DESIGNATION,
060       SITE-MANAGER.
070  SORT R1 ON SITE-NUMBER
080  PRINT R1
```

This procedure violates the sort-key rule. The LINE statements in this procedure dictate that a data instance be constructed:

1. After each retrieval of a COMPANY-REC record (LINE statement at line 030 is satisfied)

2. After each retrieval of a SITE-REC record (LINE statement at line 050 is satisfied)

Note that immediately following the retrieval of a COMPANY-REC record, however, the sort-key SITE-NUMBER is not available. Therefore, the SORT statement is invalid.

The above procedure could be made valid by adding to the LINE statement labeled L1 a data item that lies in the SITE-REC record or in subordinate records. This will cause the construction of a data entry to be postponed until a SITE-REC record has been retrieved.

20-38. Study the procedure below and determine whether the SORT statement at line 050 is valid.

```
010 QUERY PRC606/COMPADFO
020 REPORT R1 ON PRINTER
030 L1. LINE COMPANY-NAME, PRESIDENT,
040           SITE-MANAGER.
050 SORT R1 ON SITE-NUMBER
060 PRINT R1
```

●●●

```
030      TYPING-SPEED-WPM
040 PRINT ON PRINTER SITE-DESIGNATION,
050      SITE-MANAGER, CLERICAL-NAME, TECHNICAL-NAME
060 END
```

Is the SORT statement valid?

●●●

20-43. The SORT ON... statement has the following format:

```
SORT ON sort-key-1    [IN { ASC | ASCENDING | DSC | DESCENDING }    { ORDER | SEQUENCE } ]

      [,sort-key-n    [IN { ASC | ASCENDING | DSC | DESCENDING } { ORDER | SEQUENCE } ] ...]
```

The SORT ON... statement is most commonly used in conjunction with the PRINT ON... statement. It specifies that each data instance is to consist of all identifiers referenced in a PRINT statement, in addition to any sort-keys specifed in the SORT statement. The sort-keys may be specified as identifiers in a PRINT statement, although they need not be.

20-44. The second form of the QUERY SORT statement is the SORT report-name ON... statement, the format of which is as follows:

```
SORT report-name
     ON sort-key-1    [IN { ASC | ASCENDING | DSC | DESCENDING }    { ORDER | SEQUENCE } ]

      [,sort-key-n    [IN { ASC | ASCENDING | DSC | DESCENDING } { ORDER | SEQUENCE } ]...]
```

This form of the SORT statement is usually used in conjunction with the PRINT label and PRINT report-name statements. It specifies that each data instance is to consist of all of the items specified in the report body of the named report, as well as the sort-keys specified in the SORT statement. The sort-keys may also appear in the report body, although this is not required.

20-45. The third form of the QUERY SORT statement is the SORT identifier *ON... statement*, which has the following format:

```
SORT identifier [,identifier]...]
     ON sort-key-1 . [IN { ASC       } { ORDER    } ]
                         { ASCENDING } { SEQUENCE }
                         { DSC       }
                         { DESCENDING }

     [,sort-key-n    [IN { ASC       } { ORDER    } ]...]
                         { ASCENDING } { SEQUENCE }
                         { DSC       }
                         { DESCENDING }
```

When this form is used, each data instance will consist of all of the identifiers and sort-keys specified. Each sort-key may or may not be one of the specified identifiers.

The SORT identifier statement is not used as much as the other two forms of the SORT statement, and it has less flexibility (since the addition of another data item to a report may necessitate the addition of that item to the list of identifiers in the SORT statement). One use for the SORT identifier statement is the case in which multiple REPORT statements are present in the procedure (since the SORT report-name statement can reference only one report body).

20-46. In the remaining frames of this Lesson, we will complete our discussion of QUERY by looking at the statements we have not yet discussed. These are, in alphabetical order: COPY, LET, LIBRARY, and OPTIONS ARE.

20-47. COPY statement.

The COPY statement, when used in a QUERY procedure, functions in the same manner as it does in a WWDMS procedure. It is used to copy source text from a permanent file into the procedure that is to be executed.

The format of the COPY statement is:

COPY catalog/file-string mmmm-nnnn

where catalog/file-string identifies a line-numbered ASCII permanent file, and mmmm ≤ nnnn represent an inclusive range of line numbers from the file identified by catalog/file-string, which is to be copied into the Procedure File.

The COPY statement will be discussed further in Lesson 22.

20-48. LET statement.

The LET statement, when used in a QUERY procedure, assigns a value to a user-defined variable or data item instance. This is the same function that the LET statement serves in a WWDMS procedure.

20-49. The format of the QUERY Procedure LET statement is as follows:

$$\underline{\text{LET}}\ \text{simple-identifier}\ [\text{mask}] = \left\{\begin{array}{l}\text{identifier}\\ \text{literal}\\ \text{arithmetic-expression}\\ \text{numeric-constant}\\ \text{figurative-constant}\end{array}\right\}$$

Note that no provision is made for subscripting simple-identifier. Recall from Frame 20-2 that items referenced in a QUERY procedure cannot be subscripted.

20-50. The statistical functions COUNT and SUM cannot be used in a QUERY procedure LET statement. Thus, while the expression

```
LET $SUMX = $SUMX + X
```

is legal, the expression

```
LET $SUMX = SUM X
```

is illegal.

20-51. A LET statement which contains data items is executed after each retrieve of one of the lowest level records that must be retrieved in order to satisfy the specified data items. Those data items which are not available at the time of execution (because they lie on a different retrieval path) are set to blanks (for alphanumeric items) or zeroes (for numeric items) prior to the execution.

20-52. Ordinarily, each LET statement should contain at least one reference to a data item. A LET statement which does not contain data items is executed only once - prior to the retrieval of any data. For example, note the following procedure:

```
010  QUERY PRC 606/COMPADFO
020  LET $CNT = $CNT + 1
030  PRINT ON PRINTER $CNT PIC "ZZ9",
040       COMPANY-NAME, PRESIDENT
050  END
```

The writer of this procedure had intended to keep a count of the number of COMPANY-REC records retrieved. Note, however, how the LET statement at line 020 works. The statement is executed prior to the retrieval of any data. At that time, the variable $CNT is set to its previous value (zero) plus one, or one. The statement is never executed again. Therefore, the partial output from the procedure will be as follows:

```
CNT        COMPANY-NAME    PRESIDENT
 1         ABC COMPANY     G WASHINGTON
 1         DEF COMPANY     J ADAMS
 1         GHI COMPANY     T JEFFERSON
```

Clearly, this output is not as the user had intended. What the LET statement needs is a data item from the COMPANY-REC record to act as a "trigger", causing the LET statement to be executed each time a COMPANY-REC record is retrieved. Note, however, that we do not want the value of this data item to have any bearing on the result of the LET statement. See if you can write a LET statement that will accomplish the desired purpose before you read on.

●●●

```
010  QUERY PRC606/COMPADFO
020  LET
030  PRINT ON PRINTER $CNT PIC "ZZZ9",
040       COMPANY-NAME, SITE-MANAGER
050  END
```

Which of the following LET statements (a through e) could be used at line 020 to keep a count of the number of SITE-REC records retrieved? (More than one answer may be correct.)

a) `LET $CNT = $CNT + 1`

b) `LET $CNT = $CNT + 1 + (0 * COMP-SITE-NO)`

c) `LET $CNT = $CNT + 1 + (SITE-NUMBER - SITE-NUMBER)`

d) `LET $CNT = $CNT + (SITE-SITE-NO/SITE-SITE-NO)`

e) `LET $CNT = $CNT + 1 + (SITE-TYPE - SITE-TYPE)`

●●●

LIBRARY catalog/file-string

Note that there is no difference in syntax between the QUERY procedure LIBRARY statement and the WWDMS procedure LIBRARY statement.

20-55. OPTIONS ARE statement

The QUERY procedure OPTIONS ARE statement has the same function as its WWDMS counterpart: to specify default values for RUN command options for a procedure.

The format of the QUERY procedure OPTIONS ARE statement is as follows:

OPTIONS ARE option-1, option-2...

where:

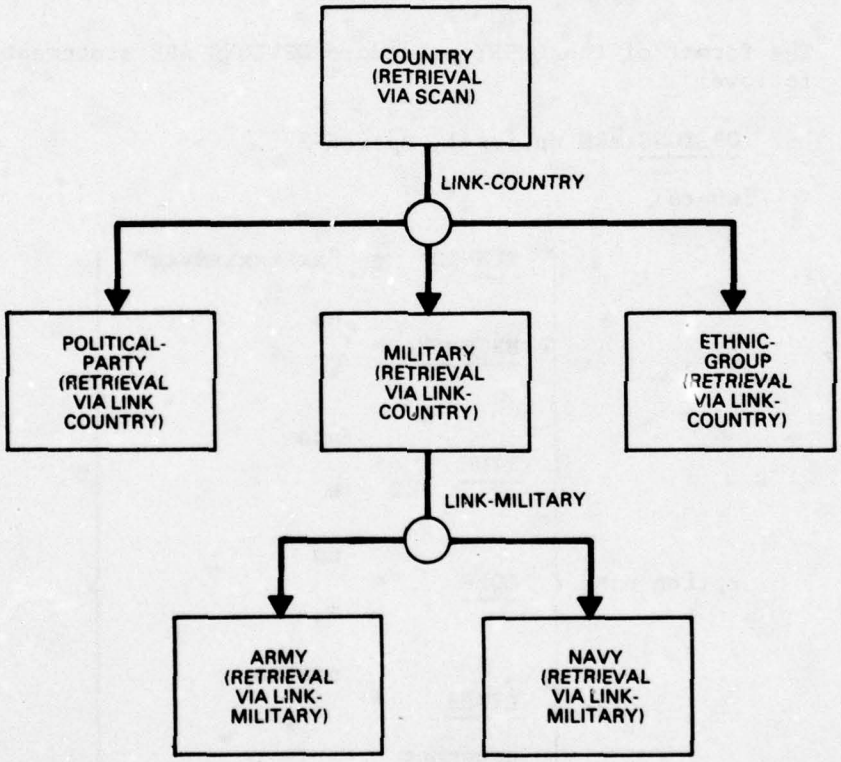
```
                 ┌ RUN-ID    =  "xxxxxxxxxxxx"  ┐
                 │                              │
                 │ URGENCY   =  nn                │
                 │              *                 │
                 │                              │
                 │ TIME      =  nnn               │
                 │              *                 │
                 │                              │
option-n:=       { CORE      =  nn                } 
                 │              *                 │
                 │                              │
                 │ LINES     =  nnnnnn            │
                 │                 *              │
                 │                              │
                 └ SCHEDULE                       ┘
```

There is one additional option which is available for use in WWDMS procedures but not in QUERY procedures. This is the TEST option, which is used to invoke the test mode feature of the Honeywell File Management Supervisor so that data base creation and update will be performed on a temporary system file instead of an actual data file. Since it is not possible to create or update data bases using a QUERY procedure, there is no need for the TEST option in the QUERY procedure language.

The RUN command options will be discussed further in Lesson 22.

20-56. TEST - LESSON 20

The following COUNTRY-FILE data structure is to be used to answer the questions in this test. A partial DDF follows the graphic representation of the structure. Assume that a default ADF is used to access the data base.




```
01  COUNTRY
      02  COUNTRY-NAME
      02  CAPITAL
      02  POPULATION

01  POLITICAL-PARTY
      02  PARTY-NAME
      02  PARTY-LEADER
      02  PER-CENT-POP

01  MILITARY
      02  CDR-IN-CHIEF
      02  MIL-NO-MEN
```

```
01  ARMY
     02  CHIEF-OF-STAFF
     02  ARMY-NO-MEN

01  NAVY
     02  CNO
     02  NAVY-NO-MEN

01  ETHNIC-GROUP
     02  GROUP-NAME
     02  GP-PERCENT
     02  LANGUAGE
```

The following procedure is to be used to answer questions 1 and 2:

```
010  QUERY PRC606/COUNDAFO
020     WHERE POPULATION GT 50000
030  PRINT ON PRINTER COUNTRY-NAME. PARTY-LEADER,
040     CDR-IN-CHIEF, CHIEF-OF-STAFF, CNO
050  END
```

1. Which other data items will be displayed whenever the item CHIEF-OF-STAFF is displayed?

2. Choose from the following list the items that could have been used in the conditional expression at line 020.

   a. COUNTRY-NAME

   b. MIL-NO-MEN

   c. GP-PERCENT

   d. CAPITAL

Refer to the procedure below to answer questions 3 through 5.

```
010  QUERY PRC606/COUNDAFO
020  REPORT R1 ON PRINTER
030     TOTAL PAGES
040     PAGE HEADING IS PG-HDG
050     PAGE FOOTING IS PG-FTG
060  PG-HDG. LINE "POLITICAL AND MILITARY REPORT" COL 30.
070          SPACE 3.
080          LINE "COUNTRY" COL 5, "POLITICAL PARTY"
085          COL 25,
090               "MILITARY LEADER" COL 45,
095               "SIZE OF ARMY" COL 65,
```

```
100              "SIZE OF NAVY" COL 80.
110          SPACE 1.
120  PG-FTG. SPACE 2.
130          LINE "PAGE" COL 30, %PAGE-NUMBER, "OF",
140               %TOTAL-PAGES.
150  DTL1.   LINE COUNTRY-NAME COL 5.
160          LINE PARTY-NAME COL 25.
170  DTL2.   LINE CDR-IN-CHIEF COL 45, ARMY-NO-MEN COL 65.
180  DTL3.   LINE NAVY-NO-MEN COL 80.
190          LINE "NAVY SUBTOTAL:" COL 45, $TOTAL-NAVY.
200  DTL4.   LINE "ETHNIC GP" COL 100, "PCT" COL 115.
210          LINE GROUP-NAME COL 100, GP-PERCENT COL 110.
220  LET $TOTAL-NAVY = $TOTAL-NAVY + NAVY-NO-MEN
230  PRINT R1
240  END
```

3. Complete the sentences below by choosing the best answers among the choices in parentheses.

   a. The LINE statement at line 190 will always be displayed immediately (before, after) the LINE statement at line (180, 200, 210).

   b. The LINE statement at line 200 will always be displayed immediately (before, after) the LINE statement at line (180, 190, 210)

4. TRUE OR FALSE:

   Since lines 150 and 160 are part of the same detail-line-set, the same event "triggers" the display of both of these lines; line 150 first, with line 160 following immediately.

5. TRUE OR FALSE:

   If we were to remove the labels from the LINE statements at lines 170 and 180 and rerun the procedure, the output would remain unchanged.

Refer to the procedure below to answer question 6.

```
010  QUERY PRC606/COUNDAFO
020  REPORT R1 ON PRINTER
030     PAGE HEADING IS PG-HDG
040  PG-HDG.  LINE "EXAMPLE OF USE OF QUERY SORT" COL 20.
050           SPACE 3.
060           LINE "COUNTRY" COL 5, "MILITARY HEAD"
080           COL 25. SPACE 1.
090  DTL1.    LINE COUNTRY-NAME COL 5, CDR-IN-CHIEF COL 25.
```

```
100  DTL2.  LINE "ARMY HEAD IS" COL 10, CHIEF-OF-STAFF
110  DTL3.  LINE "NAVY HEAD IS" COL 10, CNO
120  SORT R1 ON POPULATION DESCENDING
130  PRINT R1
140  END
```

6. Choose from the following list the items that could have been used as sort-keys.

   a. CAPITAL

   b. MIL-NO-MEN

   c. ARMY-NO-MEN

   d. NAVY-NO-MEN

●●●

5. TRUE. All LINE statements that contain data items, whether they are labled or unlabeled, are displayed following retrieval of the lowest level record along a retrieval path that must be traversed to satisfy the specified data items.

6 a. and b. Data instances are constructed at the following points:

(1) After each retrieval of a MILITARY record (satisfies the first LINE statement)

(2) After each retrieval of an ARMY record (satisfies the second LINE statement)

(3) After each retrieval of a NAVY record (satisfies the third LINE statement)

The only items that are always available whenever a data instance is to be constructed are those that lie in the COUNTRY or MILITARY records.

●●●

INDEX

	Page
ABSENT	
Absence Test	5-7
ACCEPT	
ACCEPT Statement	21-4
% ACCEPT System variable	21-5
ADFQ	
ADF Query (ADFQ)	9-9
APPLICATION DEFINITION	
ADF Query (ADFQ)	9-9
ALL ITEMS phrase	9-6
Application Definition File	2-3
Application Definition File	2-7
Application Definition File	9-1
Application Definition File user routine calls	19-4
COORDINATED ENTRIES	9-8
Data Base entry	9-1
DATA BASE IS clause	9-1
Data Base Reference	9-1
Default ADF	20-2
INPUT EDITED phrase	9-8
ITEMS ARE clause	9-5
REDEFINES phrase	9-7
REPLACES phrase	9-7
Single record entry format	14-2
Subschema	9-1
ARITHMETIC EXPRESSION	
Arithmetic expression precedence rules	7-4
ARRAY	
Control variable	10-14
DEFINE statement for arrays	10-11
Repeating data fields	8-7
Repeating data items in the Data File	10-10
Saving arrays	10-17
VARY/NEXT loop	10-15
WWDMS subscript	10-12
AUDIT	
AUDIT option (CREATE)	23-17
AUDIT option (UPDATE)	24-5
BOOLEAN FUNCTION	
Boolean function test	5-8
BREAK	
BREAK ALWAYS ON identifier CHANGE statement	16-3
BREAK FIRST ON identifier CHANGE statement	16-3
BREAK LAST ON identifier CHANGE statement	16-3
BREAK ON identifier CHANGE statement	16-2
BREAK statement	16-1

	Page
SAVE statement	16-8
Unconditional BREAK statement	16-2
CHECK	
CHECK IS clause	23-7
PICTURE validity check	23-7
%PICTURE system variable	23-7
Subroutine validity check	23-10
VALUE validity check	23-8
%VALUE system variable	23-9
CHECK COMMAND	
CHECK command	3-9
COLLATING SEQUENCE	
Collating sequence	5-2
Standard collating sequence	5-4
COLUMN	
COLUMN clause	11-7
COLUMN SPACING clause	12-4
CONDITIONAL	
Abbreviated compound conditionals	5-16
Absence Test	5-7
Boolean Function Test	5-8
Comparison on Non-numeric Items	5-3
Comparison of Numeric Items	5-3
Compound Conditional	5-12
Conditional expression	5-1
Evaluating compound conditionals	5-13
Implied subject abbreviation	5-16
Implied subject and relation abbreviation	5-16
Presence Test	5-7
QUERY Procedure Conditional expression	20-13
Relation Test	5-3
Simple conditional	5-1
Special Relational Test	5-9
Truth Table	5-12
CONTROL	
CONTROL segment	13-37
CONTROL statement	13-37
CONTROL statement	16-20
ENABLE statement	13-37
ENABLE statement	16-21
ENABLING subclause of the COUNT or SUM function	13-37
ENABLING subclause of the COUNT or SUM function	16-21
COORDINATED ENTRIES	
COORDINATED ENTRIES	9-8
COPY	
COPY Statement	22-15

	Page
QUERY Procedure COPY statement	20-33
CORE	
CORE option	22-13
COUNT	
COUNT function	7-7
COUNT function	10-4
ENABLING subclause of the COUNT function	13-37
ENABLING subclause of the COUNT function	16-21
COVER PAGE	
COVER PAGE clause	12-11
CREATE	
AUDIT option	23-17
CHECK errors	23-6
CREATE statement	23-1
EQUATE clause	23-3
Implicit CREATE procedure	23-11
INHIBIT clause	23-3
PLACING clause	23-2
REJECT statement	23-14
SEQUENCE errors	23-6
STORE statement	23-13
TEST option	23-18
Transaction data base	23-1
Transaction Reject File	23-5
WHEN...CHECK ERROR statement	23-15
WHEN...REJECT ERROR statement	23-16
WHEN...TRANSACTION SEQUENCE ERROR statement	23-14
WHERE/WITH clause	23-3
CRT	
FOR CRT clause	4-3
DAC	
PRINT ON DAC option	4-3
DATA BASE	
ADF DATA-BASE IS clause	9-1
Data base contents	1-2
WWDMS data base	1-1
WWDMS data base	2-1
DATA BASE ADMINISTRATOR	
Data Base Administrator	1-1
Data Base Administrator	8-3
DATA BASE REFERENCE	
Data Base Reference	2-2
Data Base Reference	9-1
DATA DEFINITION	
CHECK IS clause	23-7
Data Definition File	2-2

	Page
Data Definition File	2-7
Data Definition File	8-1
Data Definition File user routine calls	19-3
Data group definition	8-1
Data item definition	8-2
Data item PICTURE clause	8-6
Data Privacy	8-7
DECODING clause	8-8
DECODING clause	19-3
ENCODING clause	8-8
ENCODING clause	19-3
JUSTIFIED RIGHT clause	8-8
OCCURS clause	8-7
Record definition	8-1
Schema name	8-1
Set	8-5
SYNCHRONIZED clause	8-8
USAGE clause	8-8
DATA FILE	
Data File	2-1
DECODING	
DECODING clause	8-8
DECODING clause	19-3
DEFAULT APPLICATION DEFINITION	
Default ADF	20-2
DEFINE	
DEFINE statement for arrays	10-11
DEFINE statement for user-variables	10-2
DELETE	
DELETE statement	24-16
DFAULT	
DFAULT command	25-4
DIRECT	
DIRECT statement	22-20
DIRECT ACCESS	
ACCEPT statement	21-4
DISPLAY statement	21-6
PRINT/REPORT ON DAC option	21-2
RUNP and Direct Access output	21-1
WWDMS Terminal Executive Batch Processor (DMTEX)	21-2
DIRECTORY	
Directory File	2-2
Directory File	2-7
WWDIRUPD (WWDMS Directory File Update Subsystem)	25-7
DISPLAY	
DISPLAY statement	21-6

	Page
DJST	
DJST command	3-10
DUPLICATE	
DUPLICATE clause	11-2
ENABLE	
ENABLE statement	13-37
ENABLE statement	16-21
ENABLING subclause of the COUNT and SUM functions	13-37
ENABLING subclause of the COUNT and SUM functions	16-21
ENCODING	
ENCODING clause	8-8
ENCODING clause	19-3
END	
END statement	4-2
QUERY Procedure END statement	20-19
ENTRY	
Data base entry	9-1
Entry	2-3
Entry buffer allocation	10-4
ENTRY...IS ENTRY (super entry)	14-19
ITEMS ARE clause	14-18
Multiple record entry	14-9
OPTIONAL records	14-15
Primary access path	14-3
Primary entry	14-1
Single record entry format	14-2
THRU clause	14-6
THRU entry	14-1
VIA clause	14-3
WITH...LEVELS clause	14-18
ERROR	
CHECK errors	23-6
SEQUENCE errors	23-6
FILE	
File	1-3
FOR CRT clause	4-3
FOR PRINTER clause	4-3
FOR TTY clause	4-3
Object file	1-3
PRINT ON FILE option	4-3
Source file	1-3
FLOW OF CONTROL	
Flow of control within SORT procedures	6-5
FOOTING	
CONTROL FOOTING clause	12-24
Page footing	4-3

	Page
PAGE FOOTING clause	12-11
Statistical Functions clause	12-25
FUNCTION	
COUNT function	7-7
COUNT function	10-4
ENABLING subclause of the COUNT and SUM functions	13-37
ENABLING subclause of the COUNT and SUM functions	16-21
Statistical Functions clause	12-25
SUM function	7-5
SUM function	10-4
User functions	19-2
GOSUB	
GOSUB statement	16-13
GO TO	
GO TO statement	15-4
GROUP	
Data group	1-2
Data group definition	8-1
GROUP INDICATE	
GROUP INDICATE clause	11-18
SUPPRESS clause	11-18
HEADING	
CONTROL HEADING clause	12-23
Page heading	4-3
PAGE HEADING clause	12-11
HOLD	
HOLD statement	22-18
HOLD WITHIN label	17-5
ID	
ID option	22-10
IDENTIFIER	
Alphanumeric identifier mask	5-2
IDS FILES	
CALC (calculated) record	C-10
Data field modification	24-33
Double-chaining	C-17
IDS control field modification	24-35
IDS Data File maintenance	24-30
IDS page	C-8
IDS record deletion	24-30
IDS Subfile	C-8
Integrated Data Store (IDS) Files	C-8
LET KEY statement	24-34
New IDS record insertion	24-36
Primary Record	C-13

	Page
Secondary Record	C-12
Set (chain) ordering	C-15
IF	
ELSE clause	15-3
IF statement	15-1
THEN clause	15-3
INITIALIZE	
INITIALIZE statement	10-5
INPUT EDITED	
INPUT EDITED phrase	9-8
INSERT	
INSERT statement	24-16
INSTANCE	
Data instance	2-4
INVERTED FILES	
Inverted Files	C-7
INVOKE	
INVOKE statement	4-1
ISP FILES	
Indexed Sequential Processor (ISP) Files	C-3
ITEM	
Data item definition	8-2
Elementary data item	1-2
ITEMS ARE	
ADF ITEMS ARE clause	9-5
ALL ITEMS phrase	9-6
INPUT EDITED phrase	9-8
ITEMS ARE clause	14-18
REDEFINES phrase	9-7
REPLACES phrase	9-7
JUSTIFIED	
DDF JUSTIFIED RIGHT clause	8-8
JUSTIFIED clause	11-15
KEEP	
KEEP statement	22-16
RETRIEVE...SUBSET statement	22-17
Subset	22-16
LABEL	
Event label	13-7
Labels	13-5
Procedure control label	13-6
Report-packet label	13-8
Routine label	13-8
LET	
LET KEY statement	24-34

	Page
LET statement	7-2
QUERY Procedure LET statement	20-34
LIBRARY	
FROM subclause	19-8
Library identification in a Directory File	19-8
LIBRARY statement	19-8
QUERY Procedure LIBRARY statement	20-36
WWDMS default system library	19-5
LINE	
LINE statement	12-9
QUERY Procedure LINE statement	20-23
LINES	
LINES option	22-13
MAINTENANCE	
Advantages and disadvantages to WWDMS procedure maintenance	24-1
MASK	
Alphanumeric identifier mask	5-2
Masked variables	10-3
MATCH	
UPDATE statement MATCH clause	24-4
MOVE	
MOVE statement	10-5
NO PAGE NUMBER	
NO PAGE NUMBER clause	12-7
NOTATION	
Standard syntax notation	3-1
NUMBER OF PAGES	
NUMBER OF PAGES clause	12-5
OCCURS	
OCCURS clause	8-7
OPTION	
CORE option	22-13
ID option	22-10
LINES option	22-13
OPTIONS ARE statement	22-14
QUERY Procedure OPTIONS ARE statement	20-37
RUN command options	22-10
RUN-ID option	22-11
SCHEDULE option	22-12
TEST option	22-14
TIME option	22-13
URGENCY option	22-12
OPTIONAL	
OPTIONAL records in an entry	14-15

	Page
PAGE LENGTH	
Default page length settings	12-4
PAGE LENGTH clause	12-4
PAGE WIDTH	
Default page length settings	12-4
PAGE WIDTH clause	12-5
PARAMETER	
Default value source parameter	22-2
Object parameter	3-7
Object parameter	22-6
Optional value source parameter	22-1
Parameterized Procedure	3-7
Required value source parameter	22-2
Source parameter	3-7
Source parameter	22-1
PERFORM	
PERFORM Subsystem	25-9
PHASE	
Event statement processing phase	17-1
Subphase	17-2
PICTURE	
Data item PICTURE clause	8-6
Output PICTURE clause	11-8
System variable default output PICTUREs	11-12
PLACING	
CREATE statement PLACING clause	23-2
PRESENT	
Presence Test	5-7
PRINT	
DUPLICATE clause	11-2
FOR CRT clause	4-3
FOR PRINTER clause	4-3
FOR TTY clause	4-3
PRINT label statement	12-15
Printing an entire entry	11-24
PRINT ON DAC option	4-3
PRINT ON DAC option	21-2
PRINT ON FILE option	4-3
PRINT ON PRINTER option	4-3
PRINT ON REMOTE option	4-3
PRINT ON... statement	11-1
PRINT statement	4-3
QUERY Procedure PRINT statement	20-15
Report-element-descriptors	11-3
PRINTER	
FOR PRINTER clause	4-3

	Page
PRINT ON PRINTER option	4-3
PRIOR	
PRIOR clause	10-9
PRIVACY	
Data Privacy	8-7
Privacy File	2-4
Privacy lock	2-4
PROCEDURE	
WWDMS Procedure	1-1
Procedure Files	2-1
QUERY	
Conditional Expression	20-13
COPY statement	20-33
Default APPLICATION DEFINITION	20-2
END statement	20-19
LET statement	20-34
LIBRARY statement	20-36
LINE statement	20-23
OPTIONS ARE statement	20-37
PRINT statement	20-15
QUERY statement	20-8
REPORT statement	20-20
SORT statement	20-25
SPACE statement	20-25
READ	
READ statement	18-6
RECORD	
Record	1-2
Record definition	8-1
REDEFINES	
REDEFINES phrase	9-7
REDIRECT	
REDIRECT command	4-4
REJECT	
REJECT statement	23-14
REJECT statement	24-17
RELATION	
RELATION Test	5-3
REMARK	
REMARK statement	4-9
REMOTE	
PRINT ON REMOTE option	4-3
REPLACES	
REPLACES phrase	9-7
REPORT	
COLUMN SPACING clause	12-4

	Page
CONTROL FOOTING clause	12-24
CONTROL HEADING clause	12-23
COVER PAGE clause	12-11
LINE statement	12-9
NO PAGE NUMBER clause	12-7
NUMBER OF PAGES clause	12-5
PAGE FOOTING clause	12-11
PAGE HEADING clause	12-11
PAGE LENGTH clause	12-4
PAGE WIDTH clause	12-5
PRINT label statement	12-15
QUERY Procedure REPORT statement	20-20
Report body	12-8
REPORT ON DAC option	21-2
REPORT-packet	12-9
REPORT statement	12-1
Required clauses of a REPORT statement	12-2
SCC IS clause	12-4
SPACE statement	12-9
STARTING PAGE NUMBER clause	12-6
TITLE clause in a report body	12-21
TOTAL PAGES clause	12-6
REPORT ELEMENT DESCRIPTOR	
COLUMN clause	11-7
GROUP INDICATE clause	11-18
JUSTIFIED clause	11-15
PICTURE clause	11-8
Report element descriptor	4-3
Report element descriptor	11-3
Statistical Functions clause	12-25
SUPPRESS clause	11-18
TEXT clause	11-23
TITLE clause	11-3
RESTORE	
RESTORE statement	24-17
RETRIEVE	
How to RETRIEVE a simple entry	14-4
How to RETRIEVE multirecord entries	14-10
IDS Data File maintenance	24-30
Multiple entry RETRIEVE statement	13-17
Nested retrieval vs. OPTIONAL entry retrieval	14-17
Nested RETRIEVE...FOR UPDATE statements	24-28
Nested RETRIEVEs	14-6
ONLY nnn ENTRIES clause	17-4
Retrieval of data in a procedure	2-4
RETRIEVE...FOR UPDATE statement	24-24

	Page
RETRIEVE statement	4-2
RETRIEVE statement	5-1
RETRIEVE...SUBSET statement	22-17
Retrieving an entry having an ITEMS ARE clause	14-18
TEST option (RETRIEVE...FOR UPDATE)	24-20
RETURN	
RETURN statement	15-7
ROUTINE	
GOSUB statement	16-13
Nested ROUTINE segments	16-15
ROUTINE segment	13-3
ROUTINE statement	16-12
RUN	
DFAULT command	25-4
IDENT prompt	3-5
NOGO option of the SAVE ON clause	22-9
RUN command	3-3
RUN command	3-6
RUN command continuation character (&)	3-8
RUN command options	22-10
RUNF command	3-6
Run identification (RUN-ID) prompt	3-5
RUNJ command	3-7
RUNP and Direct Access output	21-1
RUNP command	3-7
RUNS command	3-6
SAVE ON clause	3-4
RUN-ID	
RUN-ID option	22-11
SAVE	
PRIOR clause	10-9
SAVE statement	10-8
SAVE statement	16-8
Saving arrays	10-17
SCC	
SCC IS clause	12-4
SCHEDULE	
SCHEDULE option	22-12
SCHEMA	
Schema name	8-1
Subschema	9-1
SEGMENT	
CONTROL segment	13-37
Explicit segment	13-1
Implicit WHEN segment	13-24
Main segment	13-1

	Page
ROUTINE segment	13-3
WHEN segment	13-2
SEQUENTIAL FILES	
Sequential Files	C-1
SET	
Set	8-5
Set detail (subordinate)	8-5
Set master (dominant)	8-5
SORT	
Flow of control within SORT procedures	6-5
QUERY Procedure SORT statement	20-25
Sort key	6-1
SORT statement	6-1
SORT WITHIN label	17-5
SPACE	
QUERY Procedure SPACE statement	20-25
SPACE statement	12-9
SPECIAL RELATIONAL	
Initial Mode	5-10
Special Relational Test	5-9
String Mode	5-10
Word Mode	5-10
STARTING PAGE NUMBER	
STARTING PAGE NUMBER clause	12-6
STATEMENT	
End-of-statement delimiter (.)	4-1
STATISTICAL FUNCTIONS	
Statistical Functions clause	12-25
STATUS	
Status monitoring	3-10
STOP	
STOP statement	17-1
STORE	
STORE statement	23-13
STORE statement	24-17
STRUCTURES	
Four types of Data File structures	C-1
Hierarchical relationships	C-2
Indexed Sequential Processor (ISP) Files	C-3
Integrated Data Store (IDS) Files	C-8
Inverted Files	C-7
Sequential Files	C-1
SUBROUTINE	
Subroutine Library File	2-5
User subroutines	19-1

INDEX (Continued)

	Page
SUBSCRIPT	
WWDMS subscript	10-12
SUBSET	
RETRIEVE...SUBSET statement	22-17
Subset	22-16
SUM	
ENABLING subclause of the SUM function	13-38
ENABLING subclause of the SUM function	16-21
SUM function	7-5
SUM function	10-4
SUPPRESS	
GROUP INDICATE clause	11-18
SUPPRESS clause	11-18
SYNCHRONIZED	
SYNCHRONIZED clause	8-8
TABLE LOOKUP	
TABLE LOOKUP clause	19-5
User table lookup routines	19-2
TELETYPE	
FOR TTY clause	4-3
TERMINATE	
TERMINATE statement	17-1
TEST	
TEST option	22-14
TEST option	23-18
TEST option	24-20
TEXT	
TEXT clause	11-23
THRU	
THRU clause	14-6
TIME	
TIME option	22-13
TITLE	
TITLE clause	11-3
TITLE clause in a report body	12-21
TOTAL PAGES	
TOTAL PAGES clause	12-6
TRANSACTION	
Transaction data base	23-1
TRANSACTION REJECT FILE	
Transaction Reject File	23-5
Transaction Reject File	24-9
UPDATE	
AUDIT option	24-5
DELETE statement	24-16
EQUATE clause	24-4

	Page
Explicit UPDATE procedure	24-13
Implicit UPDATE procedure	24-10
INHIBIT clause	24-5
INSERT statement	24-16
MATCH clause	24-4
MATCHED CONDITION	24-6
REJECT statement	24-17
RESTORE statement	24-17
STORE statement	24-17
TEST option	24-20
Transaction Reject File	24-9
UNMATCHED MASTER condition	24-6
UNMATCHED TRANSACTION condition	24-6
UPDATE...CREATING statement	24-21
UPDATE statement	24-2
WHEN...CHECK ERROR statement	24-15
WHEN...END OF MASTER FILE statement	24-15
WHEN...END OF TRANSACTION FILE statement	24-15
WHEN...ENTRY statement	24-14
WHEN...MASTER SEQUENCE ERROR statement	24-15
WHEN...MATCHED statement	24-14
WHEN...REJECT ERROR statement	24-15
WHEN...TRANSACTION SEQUENCE ERROR statement	24-15
WHEN...UNMATCHED MASTER statement	24-14
WHEN...UNMATCHED TRANSACTION statement	24-14
WHERE/WITH clause	24-4
URGENCY	
URGENCY option	22-12
USAGE	
USAGE clause	8-8
USE	
USE clause	19-5
USER	
WWDMS User	1-1
USER PROFILE SUBSYSTEM	
User Profile Subsystem	2-9
USER ROUTINE	
Application Definition File user routine calls	19-4
Data Definition File user routine calls	19-3
Library identification in a Directory File	19-8
LIBRARY statement	19-8
Modifying a user routine library with WWLI	D-33
PERFORM subsystem LIBRARY program	D-11
PERFORM subsystem WWTLU program	D-30
Procedural Reference Name	D-3
Procedure user routine calls	19-5

	Page
Subroutine Link Name	D-4
TABLE LOOKUP clause	19-5
USE clause	19-5
User functions	19-2
User Routine Library File	2-5
User Routine Library File structure	D-3
User routines	19-1
User subroutines	19-1
User table lookup routines	19-2
WWDMS default system library	19-5
WWLI (WWDMS Subroutine Library Editor)	25-5
VARIABLE	
CONTROL variable	10-14
Default size of user-variables	7-3
Default user-variable attributes	10-1
DEFINE statement for user-variables	10-2
Masked variables	10-3
%PICTURE system-variable	23-7
System-variables	7-1
System-variable default output PICTUREs	11-12
Truncation rules	10-3
User-variables	7-1
%VALUE system-variable	23-9
VARY/NEXT	
VARY/NEXT loop	10-15
VIA	
VIA clause	14-3
WHEN	
Entry-name ENTRY clause	13-16
Implicit WHEN segment	13-24
WHEN...CHECK ERROR statement	23-15
WHEN...CHECK ERROR statement	24-15
WHEN...EMPTY statement	13-16
WHEN...END OF MASTER FILE statement	24-15
WHEN...END OF TRANSACTION FILE statement	24-15
WHEN...END statement	13-15
WHEN...ENTRY statement	24-14
WHEN label statement	13-11
WHEN...MASTER SEQUENCE ERROR statement	24-15
WHEN...MATCHED statement	24-14
WHEN...PAGE BREAK statement	13-20
WHEN...REJECT ERROR statement	23-16
WHEN...REJECT ERROR statement	24-15
WHEN segment	13-2
WHEN statement	13-11
WHEN...TRANSACTION SEQUENCE ERROR statement	23-14

	Page
WHEN...TRANSACTION SEQUENCE ERROR statement	24-15
WHEN...UNMATCHED MASTER statement	24-14
WHEN...UNMATCHED TRANSACTION statement	24-14
WHERE/WITH clause	13-14
WRAP-UP statement	13-36
WISP	
Comparison of WWBCL and WISP	B-1
WISP (WWDMS Interface Software Package)	B-1
WITH. .LEVELS	
WITH...LEVELS clause	14-18
WRAP-UP	
WRAP-UP statement	13-36
WRITE	
PRINT vs. WRITE statement output	18-4
TO CARDS option	18-3
TO catalog/file-string option	18-3
WRITE statement	18-2
WWBCL	
Comparison of WWBCL and WISP	B-1
WWBCL (WWDMS Basic Control Language)	A-1
WWDIRUPD	
WWDIRUPD (WWDMS Directory File Update Subsystem)	25-7
WWLI	
Modifying a user routine library with WWLI	D-33
WWLI (WWDMS Subroutine Library Editor)	25-5
WWTUT	
WWTUT (WWDMS Tutorial Subsystem)	25-1