

Referenced Example Programs

This chapter contains additional example programs that are referenced in the Natural statements and system variables reference documentation. All these examples are contained in the library SYSEXSYN.

Note:

Generally, the example programs shown in the statement descriptions are written in structured mode. For statements where the reporting-mode syntax differs considerably from the structured-mode syntax, references to equivalent reporting-mode examples are also provided. The example programs are available in source-code form in the Natural library SYSEXSYN. Further example programs of using Natural statements are documented in the section *Referenced Example Programs* in the *Programming Guide*. These example programs are provided in the Natural library SYSEXPB. Please ask your Natural administrator about the availability of these libraries at your site. The example programs use data from the files EMPLOYEES and VEHICLES, which are supplied by Software AG for demonstration purposes.

The example programs listed here are referenced in the descriptions of the following statements or system variables:

- ASSIGN
- AT BREAK
- AT END OF DATA
- AT END OF PAGE
- AT START OF DATA
- AT TOP OF PAGE
- DEFINE SUBROUTINE
- FIND
- FOR
- HISTOGRAM
- IF
- PERFORM BREAK PROCESSING
- READ
- REPEAT
- SORT
- STORE
- UPDATE

- Example Programs for System Variables

ASSIGN

The following example is referenced in the ASSIGN/COMPUTE statement description:

ASGEX1R - ASSIGN (reporting mode)

```

** Example 'ASGEX1R': ASSIGN (reporting mode)
*****
RESET #A (N3)
      #B (A6)
      #C (N0.3)
      #D (N0.5)
      #E (N1.3)
      #F (N5)
      #G (A25)
      #H (A3/1:3)
*
#A = 5                                WRITE NOTITLE '=' #A
#B = 'ABC'                            WRITE '=' #B
#C = .45                               WRITE '=' #C
#D = #E = -0.12345                    WRITE '=' #D / '=' #E
ASSIGN ROUNDED #F = 199.999           WRITE '=' #F
#G = 'HELLO'                          WRITE '=' #G
*
#H (1) = 'UVW'
#H (3) = 'XYZ'                        WRITE '=' #H (1:3)
*
END

```

Output of Program AEDEX1R:

```

#A:      5
#B:     ABC
#C:     .450
#D:    -.12345
#E:   -0.123
#F:      200
#G:   HELLO
#H:   UVW      XYZ

```

AT BREAK

The following examples are referenced in the AT BREAK statement description:

ATBEX1R - AT BREAK (reporting mode)

```

** Example 'ATBEX1R': AT BREAK (reporting mode)
*****
*
LIMIT 10
READ EMPLOYEES BY CITY
  AT BREAK OF CITY DO
    SKIP 1
  DOEND

```

```

/*
  DISPLAY NOTITLE CITY (IS=ON) COUNTRY (IS=ON) NAME
LOOP
END

```

Output of Program ATBEX1R:

CITY	COUNTRY	NAME
AIKEN	USA	SENKO
AIX EN OTHE	F	GODEFROY
AJACCIO		CANALE
ALBERTSLUND	DK	PLOUG
ALBUQUERQUE	USA	HAMMOND ROLLING FREEMAN LINCOLN
ALFRETON	UK	GOLDBERG
ALICANTE	E	GOMEZ

ATBEX5R - AT BREAK statement with multiple break levels (reporting mode)

```

** Example 'ATBEX5R': AT BREAK (multiple break levels) (reporting mode)
*****
RESET LEAVE-DUE-L (N4)
*
LIMIT 5
FIND EMPLOYEES WITH CITY = 'PHILADELPHIA' OR = 'PITTSBURGH'
      SORTED BY CITY DEPT
      MOVE LEAVE-DUE TO LEAVE-DUE-L
      DISPLAY CITY (IS=ON) DEPT (IS=ON) NAME LEAVE-DUE-L
      AT BREAK OF DEPT
        WRITE NOTITLE /
          T*DEPT OLD(DEPT) T*LEAVE-DUE-L SUM(LEAVE-DUE-L) /
      AT BREAK OF CITY
        WRITE NOTITLE
          T*CITY OLD(CITY) T*LEAVE-DUE-L SUM(LEAVE-DUE-L) //
LOOP
*
END

```

Output of Program ATBEX5R:

CITY	DEPARTMENT CODE	NAME	LEAVE-DUE-L
PHILADELPHIA	MGMT30	WOLF-TERROINE	11
		MACKARNESS	27
	MGMT30		38
	TECH10	BUSH NETTLEFOLDS	39 24

	TECH10		63
PHILADELPHIA			101
PITTSBURGH	MGMT10	FLETCHER	34
	MGMT10		34
PITTSBURGH			34

AT END OF DATA

The following example is referenced in the AT END OF DATA statement description:

AEDEX1R - AT END OF DATA (reporting mode)

```

** Example 'AEDEX1R': AT END OF DATA (reporting mode)
*****
LIMIT 5
EMP. FIND EMPLOYEES WITH CITY = 'STUTT GART'
  IF NO RECORDS FOUND
    ENTER
  DISPLAY PERSONNEL-ID NAME FIRST-NAME
    SALARY (1) CURR-CODE (1)
  /*
  AT END OF DATA DO
    IF *COUNTER (EMP.) = 0 DO
      WRITE 'NO RECORDS FOUND'
      ESCAPE BOTTOM
    DOEND
    WRITE NOTITLE / 'SALARY STATISTICS:'
      / 7X 'MAXIMUM:' MAX(SALARY(1)) CURR-CODE (1)
      / 7X 'MINIMUM:' MIN(SALARY(1)) CURR-CODE (1)
      / 7X 'AVERAGE:' AVER(SALARY(1)) CURR-CODE (1)
  DOEND
LOOP
END

```

Output of Program AEDEX1R:

PERSONNEL ID	NAME	FIRST-NAME	ANNUAL SALARY	CURRENCY CODE

11100328	BERGHAUS	ROSE	70800	DM
11100329	BARTHEL	PETER	42000	DM
11300313	AECKERLE	SUSANNE	55200	DM
11300316	KANTE	GABRIELE	61200	DM
11500304	KLUGE	ELKE	49200	DM
SALARY STATISTICS:				
	MAXIMUM:	70800	DM	
	MINIMUM:	42000	DM	
	AVERAGE:	55680	DM	

AT END OF PAGE

The following example is referenced in the AT END OF PAGE statement description:

AEPEX1R - AT END OF PAGE (reporting mode)

```
** Example 'AEPEX1R': AT END OF PAGE (reporting mode)
*****
FORMAT PS=10
LIMIT 10
READ EMPLOYEES BY PERSONNEL-ID FROM '20017000'
  DISPLAY NOTITLE GIVE SYSTEM FUNCTIONS
    NAME JOB-TITLE 'SALARY' SALARY(1) CURR-CODE (1)
  /*
  AT END OF PAGE DO
    WRITE / 28T 'AVERAGE SALARY: ...' AVER(SALARY(1)) CURR-CODE (1)
  DOEND
  /*
LOOP
END
```

Output of Program AEPEX1R:

NAME	CURRENT POSITION	SALARY	CURRENCY CODE
CREMER	ANALYST	34000	USD
MARKUSH	TRAINEE	22000	USD
GEE	MANAGER	39500	USD
KUNEY	DBA	40200	USD
NEEDHAM	PROGRAMMER	32500	USD
JACKSON	PROGRAMMER	33000	USD
AVERAGE SALARY: ...		33533	USD

AT START OF DATA

The following example is referenced in the AT START OF DATA statement description:

ASDEX1R - AT START OF DATA (reporting mode)

```
** Example 'ASDEX1R': AT START OF DATA (reporting mode)
*****
RESET #CITY (A20) #CNTL (A1)
*
REPEAT
  INPUT 'ENTER VALUE FOR CITY' #CITY
  /*
  IF #CITY = ' ' OR= 'END' DO
    STOP
  DOEND
  FIND EMPLOYEES WITH CITY = #CITY
  IF NO RECORDS FOUND DO
    WRITE NOTITLE NOHDR 'NO RECORDS FOUND'
    ESCAPE
  DOEND
  /*
  AT START OF DATA DO
```

```

INPUT (AD=0) 'RECORDS FOUND' *NUMBER //
                'ENTER ''D'' TO DISPLAY RECORDS' #CNTL (AD=A)
IF #CNTL NE 'D' DO
    ESCAPE BOTTOM
DOEND
DOEND
/*
DISPLAY NAME FIRST-NAME
LOOP
LOOP
END

```

Output of Program ASDEX1R:

ENTER VALUE FOR CITY **PARIS**

After entering and confirming city name:

```

RECORDS FOUND          26

ENTER 'D' TO DISPLAY RECORDS D

```

After entering and confirming D:

NAME	FIRST-NAME
MAIZIERE	ELISABETH
MARX	JEAN-MARIE
REIGNARD	JACQUELINE
RENAUD	MICHEL
REMOUE	GERMAINE
LAVENDA	SALOMON
BROUSSE	GUY
GIORDA	LOUIS
SIECA	FRANCOIS
CENSIER	BERNARD
DUC	JEAN-PAUL
CAHN	RAYMOND
MAZUY	ROBERT
FAURIE	HENRI
VALLY	ALAIN
BRETON	JEAN-MARIE
GIGLEUX	JACQUES
KORAB-BRZOZOWSKI	BOGDAN
XOLIN	CHRISTIAN
LEGRIS	ROGER
VVVV	

AT TOP OF PAGE

The following example is referenced in the AT TOP OF PAGE statement description:

ATPEX1R - AT TOP OF PAGE (reporting mode)

```

** Example 'ATPEX1R': AT TOP OF PAGE (reporting mode)
*****
*
FORMAT PS=15
LIMIT 15

```

```

*
READ EMPLOYEES BY NAME STARTING FROM 'L'
  DISPLAY 2X NAME 4X FIRST-NAME CITY DEPT
  WRITE TITLE UNDERLINED 'EMPLOYEE REPORT'
  WRITE TRAILER '-' (78)
/*
  AT TOP OF PAGE DO
    WRITE 'BEGINNING NAME:' NAME
  DOEND
/*
  AT END OF PAGE DO
    SKIP 1
    WRITE 'ENDING NAME: ' NAME
  DOEND
LOOP
END

```

DEFINE SUBROUTINE

The following example is referenced in the DEFINE SUBROUTINE statement description:

DSREX1R - DEFINE SUBROUTINE (reporting mode)

```

** Example 'DSREX1R': DEFINE SUBROUTINE (reporting mode)
*****
RESET #ARRAY-ALL (A300)
  #X (N2) #Y (N2)
REDEFINE #ARRAY-ALL (#ARRAY (A75/1:4))
  #ARRAY-ALL (#ALINE (A25/1:4,1:3))
*
FORMAT PS=20
LIMIT 5
*
MOVE 1 TO #X #Y
*
FIND EMPLOYEES WITH NAME = 'SMITH'
  OBTAIN ADDRESS-LINE (1:2)
/*
  MOVE NAME          TO #ALINE (#X,#Y)
  MOVE ADDRESS-LINE(1) TO #ALINE (#X+1,#Y)
  MOVE ADDRESS-LINE(2) TO #ALINE (#X+2,#Y)
  MOVE PHONE         TO #ALINE (#X+3,#Y)
  IF #Y = 3 DO
    MOVE 1 TO #Y
    PERFORM PRINT
  DOEND
  ELSE DO
    ADD 1 TO #Y
  DOEND
  AT END OF DATA DO
    PERFORM PRINT
  DOEND
LOOP
*
DEFINE SUBROUTINE PRINT
  WRITE NOTITLE (AD=OI) #ARRAY(*)
  RESET #ARRAY(*)
  SKIP 1
RETURN
*
END

```

Output of Program AEDEX1R:

SMITH ENGLANDSVEJ 222 554349	SMITH 3152 SHETLAND ROAD MILWAUKEE 877-4563	SMITH 14100 ESWORTHY RD. MONTERREY 994-2260
SMITH 5 HAWTHORN OAK BROOK 150-9351	SMITH 13002 NEW ARDEN COUR SILVER SPRING 639-8963	

FIND

The following examples are referenced in the FIND statement description:

FNDFIR - FIND statement with FIRST option (reporting mode)

```

** Example 'FNDFIR': FIND FIRST
*****
*
FIND FIRST EMPLOYEES WITH CITY = 'DERBY'
*
WRITE NOTITLE 'TOTAL RECORDS SELECTED:' *NUMBER
SKIP 2
WRITE '***FIRST PERSON SELECTED***' //
      'NAME:      ' NAME /
      'DEPARTMENT:' DEPT /
      'JOB TITLE: ' JOB-TITLE
*
END

```

Output of Program FNDFIR:

```

TOTAL RECORDS SELECTED:      141

***FIRST PERSON SELECTED***

NAME:      DEAKIN
DEPARTMENT: SALE01
JOB TITLE: SALES ACCOUNTANT

```

FNDNUM - FIND statement with NUMBER option (reporting mode)

```

** Example 'FNDNUM': FIND NUMBER
*****
RESET #BIRTH (D)
*
MOVE EDITED '19500101' TO #BIRTH (EM=YYYYMMDD)
*
FIND NUMBER EMPLOYEES WITH CITY = 'MADRID'
      WHERE BIRTH LT #BIRTH
*
WRITE NOTITLE 'TOTAL RECORDS SELECTED:      ' *NUMBER
              / 'TOTAL BORN BEFORE 1 JAN 1950: ' *COUNTER
*
END

```


Output of Program FNDNUM:

```
TOTAL RECORDS SELECTED:          41
TOTAL BORN BEFORE 1 JAN 1950:   16
```

FNDUNQ - FIND statement with UNIQUE option (reporting mode)

```
** Example 'FNDUNQ': FIND UNIQUE
*****
RESET #NAME (A20)
*
*
INPUT 'ENTER EMPLOYEE NAME: ' #NAME
IF #NAME = ' '
  STOP
*
FIND UNIQUE EMPLOYEES WITH NAME = #NAME
*
DISPLAY NOTITLE NAME FIRST-NAME JOB-TITLE
*
ON ERROR DO
  WRITE 'NAME EITHER NOT UNIQUE OR DOES NOT EXIST'
  FETCH 'FNDUNQ'
DOEND
*
END
```

Output of Program FNDUNQ:

```
ENTER EMPLOYEE NAME: HEURTEBISE
```

After entering and confirming name HEURTEBISE:

NAME	FIRST-NAME	CURRENT POSITION
HEURTEBISE	MICHEL	CONTROLEUR DE GESTION

FOR

The following example is referenced in the FOR statement description:

FOREX1R - FOR (reporting mode)

```
** Example 'FOREX1R': FOR (reporting mode)
*****
RESET #INDEX (I1)
      #ROOT (N2.7)
*
FOR #INDEX 1 TO 5
  COMPUTE #ROOT = SQRT (#INDEX)
  WRITE NOTITLE '=' #INDEX 3X '=' #ROOT
LOOP
*
SKIP 1
FOR #INDEX 1 TO 5 STEP 2
  COMPUTE #ROOT = SQRT (#INDEX)
```

```

WRITE '=' #INDEX 3X '=' #ROOT
LOOP
*
END

```

Output of Program FOREX1R:

```

#INDEX: 1 #ROOT: 1.0000000
#INDEX: 2 #ROOT: 1.4142135
#INDEX: 3 #ROOT: 1.7320508
#INDEX: 4 #ROOT: 2.0000000
#INDEX: 5 #ROOT: 2.2360679

#INDEX: 1 #ROOT: 1.0000000
#INDEX: 3 #ROOT: 1.7320508
#INDEX: 5 #ROOT: 2.2360679

```

HISTOGRAM

The following example is referenced in the HISTOGRAM statement description:

HSTEX1R - HISTOGRAM (reporting mode)

```

** Example 'HSTEX1R': HISTOGRAM (reporting mode)
*****
*
LIMIT 8
HISTOGRAM EMPLOYEES CITY STARTING FROM 'M'
  DISPLAY NOTITLE CITY
    'NUMBER OF/PERSONS' *NUMBER *COUNTER
LOOP
*
END

```

Output of Program HSTEX1R:

CITY	NUMBER OF PERSONS	CNT
MADISON	3	1
MADRID	41	2
MAILLY LE CAMP	1	3
MAMERS	1	4
MANSFIELD	4	5
MARSEILLE	2	6
MATLOCK	1	7
MELBOURNE	2	8

IF

The following example is referenced in the IF statement description:

IFEX1R - IF (reporting mode)

```

** Example 'IFEX1R': IF (reporting mode)
*****
RESET #BIRTH (D)
*
MOVE EDITED '19450101' TO #BIRTH (EM=YYYYMMDD)
SUSPEND IDENTICAL SUPPRESS
LIMIT 20
*
FND. FIND EMPLOYEES WITH CITY = 'FRANKFURT'
      SORTED BY NAME BIRTH
  IF SALARY (1) LT 40000
    WRITE NOTITLE '*****' NAME 30X 'SALARY LT 40000'
  ELSE DO
    IF BIRTH GT #BIRTH DO
      FIND VEHICLES WITH PERSONNEL-ID = PERSONNEL-ID (FND.)
      DISPLAY (IS=ON) NAME BIRTH (EM=YYYY-MM-DD)
      SALARY (1) MAKE (AL=8)
    LOOP
  DOEND
DOEND
LOOP
END

```

Output of Program IFEX1R:

NAME	DATE OF BIRTH	ANNUAL SALARY	MAKE
BAECKER	1956-01-05	74400	BMW
***** BECKER			SALARY LT 40000
BLOEMER	1979-11-07	45200	FIAT
FALTER	1954-05-23	70800	FORD
***** FALTER			SALARY LT 40000
***** GROTHE			SALARY LT 40000
***** HEILBROCK			SALARY LT 40000
***** HESCHMANN			SALARY LT 40000
HUCH	1952-09-12	67200	MERCEDES
***** KICKSTEIN			SALARY LT 40000
***** KLEENE			SALARY LT 40000
***** KRAMER			SALARY LT 40000

PERFORM BREAK PROCESSING

The following example is referenced in the PERFORM BREAK PROCESSING statement description:

PBPEX1R - PERFORM BREAK PROCESSING (reporting mode)

```

** Example 'PBPEX1R': PERFORM BREAK PROCESSING (reporting mode)
*****
RESET #LINE (N2) #INDEX (N2)
*
MOVE 1 TO #LINE
FOR #INDEX 1 TO 18
  PERFORM BREAK PROCESSING
  /*
  AT BREAK OF #INDEX /1/ DO

```

```

        WRITE NOTITLE / 'PLEASE COMPLETE LINES 1-9 ABOVE' /
        MOVE 1 TO #LINE
    DOEND
/*
    WRITE NOTITLE '_' (64) '=' #LINE
    ADD 1 TO #LINE
LOOP
END

```

Output of Program PBPEX1R:

```

_____ #LINE: 1
_____ #LINE: 2
_____ #LINE: 3
_____ #LINE: 4
_____ #LINE: 5
_____ #LINE: 6
_____ #LINE: 7
_____ #LINE: 8
_____ #LINE: 9

```

PLEASE COMPLETE LINES 1-9 ABOVE

```

_____ #LINE: 1
_____ #LINE: 2
_____ #LINE: 3
_____ #LINE: 4
_____ #LINE: 5
_____ #LINE: 6
_____ #LINE: 7
_____ #LINE: 8
_____ #LINE: 9

```

PLEASE COMPLETE LINES 1-9 ABOVE

READ

The following example is referenced in the READ statement description:

REAEX1R - READ (reporting mode)

```

** Example 'REAEX1R': READ (reporting mode)
*****
LIMIT 3
*
WRITE 'READ IN PHYSICAL SEQUENCE'
READ EMPLOYEES IN PHYSICAL SEQUENCE
    DISPLAY NOTITLE PERSONNEL-ID NAME *ISN *COUNTER
LOOP
*
WRITE / 'READ IN ISN SEQUENCE'
READ EMPLOYEES BY ISN STARTING FROM 1 ENDING AT 3
    DISPLAY PERSONNEL-ID NAME *ISN *COUNTER
LOOP
*
WRITE / 'READ IN NAME SEQUENCE'
READ EMPLOYEES BY NAME
    DISPLAY PERSONNEL-ID NAME *ISN *COUNTER
LOOP
*

```

```
WRITE / 'READ IN NAME SEQUENCE STARTING FROM 'M''
READ EMPLOYEES BY NAME STARTING FROM 'M'
  DISPLAY PERSONNEL-ID NAME *ISN *COUNTER
LOOP
*
```

Output of Program REAEX1R:

PERSONNEL ID	NAME	ISN	CNT

READ IN PHYSICAL SEQUENCE			
50005800	ADAM	1	1
50005600	MORENO	2	2
50005500	BLOND	3	3
READ IN ISN SEQUENCE			
50005800	ADAM	1	1
50005600	MORENO	2	2
50005500	BLOND	3	3
READ IN NAME SEQUENCE			
60008339	ABELLAN	478	1
30000231	ACHIESON	878	2
50005800	ADAM	1	3
READ IN NAME SEQUENCE STARTING FROM 'M'			
30008125	MACDONALD	923	1
20028700	MACKARNES	765	2
40000045	MADSEN	508	3

REPEAT

The following examples are referenced in the REPEAT statement description:

RPTEX1R - REPEAT (reporting mode)

```
** Example 'RPTEX1R': REPEAT (reporting mode)
*****
RESET #PERS-NR (A8)
*
REPEAT
  INPUT 'ENTER A PERSONNEL NUMBER:' #PERS-NR
  IF #PERS-NR = ' '
    ESCAPE BOTTOM
  FIND EMPLOYEES WITH PERSONNEL-ID = #PERS-NR
  IF NO RECORD FOUND
    REINPUT 'NO RECORD FOUND'
  DISPLAY NOTITLE NAME
LOOP
LOOP
*
```

Output of Program RPTEX1R:

ENTER A PERSONNEL NUMBER:

RPTEX2R - REPEAT with WHILE and UNTIL option (reporting mode)

```

** Example 'RPTEX2R': REPEAT (with WHILE and UNTIL option)
*****
RESET #X (I1) #Y (I1)
*
*
REPEAT WHILE #X <= 5
  ADD 1 TO #X
  WRITE NOTITLE '=' #X
LOOP
*
SKIP 3
REPEAT
  ADD 1 TO #Y
  WRITE '=' #Y
  UNTIL #Y = 6
LOOP
*
END

```

Output of Program RPTEX2R:

```

#X: 1
#X: 2
#X: 3
#X: 4
#X: 5
#X: 6

```

```

#Y: 1
#Y: 2
#Y: 3
#Y: 4
#Y: 5
#Y: 6

```

SORT

The following example is referenced in the SORT statement description:

SRTEX1R - SORT (reporting mode)

```

** Example 'SRTEX1R': SORT (reporting mode)
*****
RESET #AVG (P11) #TOTAL-TOTAL (P11) #TOTAL-SALARY (P11)
  #AVER-PERCENT (N3.2)
*
LIMIT 3
FIND EMPLOYEES WITH CITY = 'BOSTON'
  OBTAIN SALARY(1:2)
  COMPUTE #TOTAL-SALARY = SALARY (1) + SALARY (2)
  ACCEPT IF #TOTAL-SALARY GT 0
/*
  SORT BY PERSONNEL-ID USING #TOTAL-SALARY SALARY(*) CURR-CODE
  GIVE AVER(#TOTAL-SALARY)

```

```

/*
AT START OF DATA DO
  WRITE NOTITLE '*' (40)
    'AVG CUMULATIVE SALARY:' *AVER (#TOTAL-SALARY) /
  MOVE *AVER (#TOTAL-SALARY) TO #AVG
DOEND
COMPUTE ROUNDED #AVER-PERCENT = #TOTAL-SALARY / #AVG * 100
ADD #TOTAL-SALARY TO #TOTAL-TOTAL
/*
DISPLAY NOTITLE PERSONNEL-ID SALARY (1) SALARY (2)
  #TOTAL-SALARY CURR-CODE (1)
  'PERCENT/OF/AVER' #AVER-PERCENT
AT END OF DATA
  WRITE / '*' (40) 'TOTAL SALARIES PAID: ' #TOTAL-TOTAL
LOOP
*
END

```

Output of Program SRTEX1R:

PERSONNEL ID	ANNUAL SALARY	ANNUAL SALARY	#TOTAL-SALARY	CURRENCY CODE	PERCENT OF AVER

***** AVG CUMULATIVE SALARY:					44633
20000100	31000	29400	60400	USD	135.30
20019200	18000	17100	35100	USD	78.60
20020400	20000	18400	38400	USD	86.00
***** TOTAL SALARIES PAID:					133900

STORE

The following example is referenced in the STORE statement description:

STOEX1R - STORE (reporting mode)

```

** Example 'STOEX1R': STORE (reporting mode)
**
** CAUTION: Executing this example will modify the database records!
*****
RESET #PERSONNEL-ID (A8)
      #NAME (A20)
      #FIRST-NAME (A15)
      #BIRTH-D (D)
      #MAR-STAT (A1)
      #BIRTH (A8)
      #CITY (A20)
      #COUNTRY (A3)
      #CONF (A1)
*
REPEAT
  INPUT 'ENTER A PERSONNEL ID AND NAME (OR ''END'' TO END)' //
    'PERSONNEL-ID : ' #PERSONNEL-ID //
    'NAME : ' #NAME /
    'FIRST-NAME : ' #FIRST-NAME
/*
/* VALIDATE ENTERED DATA

```

```

/*
IF #PERSONNEL-ID = 'END' OR #NAME = 'END'
  STOP
IF #NAME = ' '
  REINPUT WITH TEXT 'ENTER A LAST-NAME' MARK 2 AND SOUND ALARM
IF #FIRST-NAME = ' '
  REINPUT WITH TEXT 'ENTER A FIRST-NAME' MARK 3 AND SOUND ALARM
/*
/* ENSURE PERSON IS NOT ALREADY ON FILE
/*
FIND NUMBER EMPLOYEES WITH PERSONNEL-ID = #PERSONNEL-ID
IF *NUMBER > 0
  REINPUT 'PERSON WITH SAME PERSONNEL-ID ALREADY EXISTS'
  MARK 1 AND SOUND ALARM
MOVE 'N' TO #CONF
/*
/* GET FURTHER INFORMATION
/*
INPUT
  'ADDITIONAL PERSONNEL DATA'          ////
  'PERSONNEL-ID          : ' #PERSONNEL-ID (AD=IO) /
  'NAME                  : ' #NAME          (AD=IO) /
  'FIRST-NAME           : ' #FIRST-NAME    (AD=IO) ///
  'MARITAL STATUS       : ' #MAR-STAT      /
  'DATE OF BIRTH (YYYYMMDD) : ' #BIRTH      /
  'CITY                 : ' #CITY          /
  'COUNTRY (3 CHARACTERS) : ' #COUNTRY     //
  'ADD THIS RECORD (Y/N)  : ' #CONF        (AD=M)
/*
/* ENSURE REQUIRED FIELDS CONTAIN VALID DATA
/*
IF NOT (#MAR-STAT = 'S' OR = 'M' OR = 'D' OR = 'W')
  REINPUT TEXT 'ENTER VALID MARITAL STATUS S=SINGLE ' -
  'M=MARRIED D=DIVORCED W=WIDOWED' MARK 1
IF NOT (#BIRTH = MASK(YYYYMMDD) AND #BIRTH = MASK(1582-2699))
  REINPUT TEXT 'ENTER CORRECT DATE' MARK 2
IF #CITY = ' '
  REINPUT TEXT 'ENTER A CITY NAME' MARK 3
IF #COUNTRY = ' '
  REINPUT TEXT 'ENTER A COUNTRY CODE' MARK 4
IF NOT (#CONF = 'N' OR = 'Y')
  REINPUT TEXT 'ENTER Y (YES) OR N (NO)' MARK 5
IF #CONF = 'N'
  ESCAPE TOP
/*
/* ADD THE RECORD
/*
MOVE EDITED #BIRTH TO #BIRTH-D (EM=YYYYMMDD)
/*
STORE RECORD IN EMPLOYEES
  WITH PERSONNEL-ID = #PERSONNEL-ID
    NAME           = #NAME
    FIRST-NAME     = #FIRST-NAME
    MAR-STAT       = #MAR-STAT
    BIRTH          = #BIRTH-D
    CITY           = #CITY
    COUNTRY        = #COUNTRY
END OF TRANSACTION
/*

```



```

WRITE NOTITLE 'RECORD HAS BEEN ADDED'
/*
LOOP
END

```

UPDATE

The following example is referenced in the UPDATE statement description:

UPDEX1R - UPDATE (reporting mode)

```

** Example 'UPDEX1R': UPDATE (reporting mode)
**
** CAUTION: Executing this example will modify the database records!
*****
RESET #NAME (A20)
*
INPUT 'ENTER A NAME:' #NAME (AD=M)
IF #NAME = ' '
  STOP
*
FIND EMPLOYEES WITH NAME = #NAME
  IF NO RECORDS FOUND
    REINPUT WITH 'NO RECORDS FOUND' MARK 1
  /*
  INPUT 'NAME:          ' #NAME (AD=O) /
        'FIRST NAME:' FIRST-NAME (AD=M) /
        'CITY:          ' CITY (AD=M)
  /*
  UPDATE USING SAME RECORD
  /*
  END TRANSACTION
  /*
LOOP
*
END

```

Output of Program UPDEX1R:

```
ENTER A NAME:
```

Example Programs for System Variables

The following examples are referenced in the *OCCURRENCE system variable description:

OCC1P - System Variable *OCCURRENCE

```

** Example 'OCC1P': *OCCURRENCE
*****
DEFINE DATA LOCAL
1 #N1 (N7/1:10)
1 #N2 (N7/1:10,1:10)
1 #N3 (N7/1:10,1:10,1:10)
END-DEFINE
*
CALLNAT 'OCC1P' #N1(*) #N2(1:2,1:4) #N3(1:6,1:7,1:8)
*
END

```

Subprogram OCC1N Called by Program OCC1P:

```

** Example 'OCC1N': *OCCURRENCE (called by OCC1P)
*****
DEFINE DATA
PARAMETER
1 PARM1 (N7/1:V)
1 PARM2 (N7/1:V,1:V)
1 PARM3 (N7/1:V,1:V,1:V)
LOCAL
1 #OCC2 (I4/1:2)
1 #OCC3 (I4/1:3)
1 #OCC1 (I4)
END-DEFINE
*
MOVE *OCC(PARM1) TO #OCC1
MOVE *OCC(PARM2,*) TO #OCC2(*)
MOVE *OCC(PARM3,*) TO #OCC3(*)
*
DISPLAY #OCC1 #OCC2(*) #OCC3(*)
DISPLAY *OCC(PARM1,*) *OCC(PARM2,*) *OCC(PARM3,*)
*
NEWPAGE
*
WRITE NOHDR
'Occurrences of 1. parameter:' *OCC(PARM1)
/ 'Occurrences of 1. parameter:' *OCC(PARM1,1)
/ 'Occurrences of 1. parameter:' *OCC(PARM1,*)
/ 'Occurrences of 2. parameter:' *OCC(PARM2,1) *OCC(PARM2,2)
/ 'Occurrences of 2. parameter:' *OCC(PARM2,*)
/ 'Occurrences of 3. parameter:' *OCC(PARM3,1) *OCC(PARM3,2)
/OCC(PARM3,3)
/ 'Occurrences of 3. parameter:' *OCC(PARM3,*)
*
END

```

Output of Program OCC1P - Page 1:

Page 1 05-01-18 10:21:30

#OCC1	#OCC2	#OCC3
10	2	6
	4	7
		8
10	2	6
	4	7
		8

Output of Program OCC1P - Page 2:

Page 2 05-01-18 10:21:30

Occurrences of 1. parameter:	10		
Occurrences of 1. parameter:	10		
Occurrences of 1. parameter:	10		
Occurrences of 2. parameter:	2	4	
Occurrences of 2. parameter:	2	4	
Occurrences of 3. parameter:	6	7	8
Occurrences of 3. parameter:	6	7	8

OCC2P - System Variable *OCCURRENCE

```

** Example 'OCC2P': *OCCURRENCE
*****
DEFINE DATA LOCAL
1 #N (N7/1:10)
1 #I (I4)
END-DEFINE
*
FOR #I=1 TO 10
  MOVE #I TO #N(#I)
END-FOR
*
WRITE 'Passing occurrences 1:5'
CALLNAT 'OCC2N' #N(1:5)
*
WRITE 'Passing occurrences 5:10'
CALLNAT 'OCC2N' #N(5:10)
*
END

```

Subprogram OCC2N Called by Program OCC2P:

```

** Example 'OCC2N': *OCCURRENCE (called by OCC2P)
*****
DEFINE DATA
PARAMETER
1 #ARR (N7/1:V)
LOCAL
1 I (N7)
END-DEFINE
*
FOR I=1 TO *OCC(#ARR)
  DISPLAY #ARR(I)
END-FOR
*
END

```

Output of Program OCC2P:

Page 1

05-01-18 10:33:03

Passing occurrences 1:5

1
2
3
4
5

Passing occurrences 5:10

5
6
7
8
9
10