

fuse inc.



FUSE '93

**Educational Conference and
International Users Meeting**

Atlanta, Georgia • May 23-28, 1993

THURSDAY

 **FOCUS**®

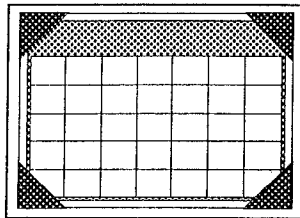
 **EDA/SQL**™

 **LEVEL5**®

Table of Contents

A FOCUSed Look at Dates.....	TH1
A Reporting Technique Report Lines: Creating Many From One or One From Many.....	TH2
Adding Printer Control To Your TABLE Requests.....	TH3
Building Lean Multi-User Applications Without SU.....	TH4
Controlling Application Development with FOCUS.....	TH5
Creating Custom Application Trace Files.....	TH6
Dynamic EMR Code Generation.....	TH7
EIS and Graphical Reporting Systems - Designing Solutions....	TH8
Explaining Explain Improving Your DB2 Table Request.....	TH9
Fast Name Searches for PC/FOCUS LAN Applications and Other Efficiency Techniques.....	TH10
FOCUS 7.0.....	TH11
FOCUS for AS/400 Advanced Communications.....	TH12
FOCUS Report Generators Description.....	TH13
Just Say "NO" to Hardcoding	TH14
LEVEL5 Application Development Tips and Techniques.....	TH15
LOOP-D-LOOP.....	TH16
MODIFY Basics - No Nonsense Basics of MODIFY.....	TH17
Next Generation Client/Server.....	TH18
Object Database and 4GLs A Paradigm Shift or a Paradigm Transition.....	TH19
PC Memory Management.....	TH20
Performance Management Using SmartMode.....	TH21

A FOCUSed Look at Dates



Larry Richman

FUSE '93
Educational Conference
May 23-28, 1993

Overview

1. Date Formats in FOCUS
 - Alphanumeric/Integer Format with a Date Edit Option
 - FOCUS Internal Date Format
2. Date Conversions: Getting From Here to There
 - Converting Alphanumeric/Integer Dates to Internal Dates
 - Converting Internal Dates to Alphanumeric/Integer Dates
 - Converting Components of Dates in Internal Date Format
 - Converting Components of Dates in Alphanumeric/Integer Format
3. Arithmetic Operations
 - Arithmetic Operations with Internal Date Format
 - Arithmetic Operations with Alphanumeric/Integer Date Format
4. Getting Today's Date and Time
 - Using System Variables
 - Using User-Written Subroutines
5. Calendar Systems
 - Converting from Gregorian to Julian
 - Converting from Julian to Gregorian
6. Number of Days this Century
 - Converting from Dates to Number of Days this Century
 - Converting from Number of Days this Century to Dates
7. Displaying the Day of the Week
8. Moving into the 21st Century
9. Date Displays

All examples in this presentation were verified using FOCUS Release 6.0

1. Date Formats in FOCUS

1. Alphanumeric or integer format with a date edit option
2. FOCUS internal date format

• Alphanumeric and Integer Format with a Date Edit Option

These fields are stored in FOCUS as simple strings of digits, which only have meaning as dates because of the format specification. For example:

<u>Format</u>	<u>Display</u>
I6MDY	05/27/93
I6YMD	93/05/27
I6DMY	27/05/93
I8DMYY	27/05/1993

The date edit options are D, M, Y, and T. The month's number (1 to 12) can be translated to the corresponding month name by adding the letter T to the format, immediately after the M. For example:

<u>Format</u>	<u>Data</u>	<u>Display</u>
I6MTDY	05/27/93	MAY 27 93
I4MTY	0593	MAY 93
I2MT	05	MAY

To sort dates in calendar order rather than alphabetic order, define a field with just the month element as I2MT. This field will display the values JAN, FEB, MAR, etc. in the correct calendar order because the sorting is based on the numeric values, not the alphabetic values.

- **FOCUS Internal Date Format**

Fields in FOCUS Internal Date Format are stored internally as four-byte binary integers representing the elapsed time since the base date December 31, 1900.

Using the FOCUS Internal Date Format, you can

- use different formats for input, sorting, and display (input 052793, sort 930527, display 27 MAY 93).
- do arithmetic with dates in different formats without the need for special date-handling routines.
- validate dates automatically in transactions.
- define and extract date components (such as, year, quarter, month, day, day of week).

Note:

- Dates set to the base date display as blanks.
- Date fields entered as blank or all zeros are accepted during validation and are interpreted as the base date.

2. Date Conversions: Getting from Here to There

- **Converting Alphanumeric/Integer Dates to Internal Dates**

To convert from A6MDY to MDY, use simple computational expressions of the form

```
COMPUTE DATE1/MDY = DATE2;
```

You can define an alphanumeric or integer date as an internal date as shown below. (HIRE_DATE was declared as A6MDY in the Master File Description, and RAISE_DATE was declared as I6YMD.)

```
DEFINE FILE EMPLOYEES  
HIRE_DATE/MDY = HIRE_DATE;  
RAISE_DATE/YMD = RAISE_DATE;  
END
```

You can also convert directly in the table request without using DEFINE as shown below.
(ORDER_DATE was declared as I6MDY in the Master File Description.)

```
TABLE FILE TEST
PRINT ORDER_DATE/MDY      AS 'AS MDY'
   ORDER_DATE/MDYY       AS 'AS MDYY'
   ORDER_DATE/YMD        AS 'AS YMD'
   ORDER_DATE/YYMD       AS 'AS YYMD'
   ORDER_DATE/MTDY       AS 'AS MTDY'
   ORDER_DATE/M-D-Y      AS 'AS M-D-Y'
END
```

Output:

AS MDY	AS MDYY	AS YMD	AS YYMD	AS MTDY	AS M-D-Y
05/27/93	05/27/1993	93/05/27	1993/05/27	MAY 27, 93	05-27-93

• Converting Internal Dates to Alphanumeric/Integer Dates

To convert from MDY to I6MDY or A6MDY:

```
DEFINE FILE TEST
DATE2/I6MDY = DATE1;
DATE3/A6MDY = DATE1;
END
TABLE FILE TEST
PRINT DATE1 DATE2 DATE3
END
```

Output:

DATE1	DATE2	DATE3
05/27/93	05/27/93	05/27/93

• **Converting Components of Dates in Internal Date Format**

To convert from MDY to YMD:

ORDER_DATE/YMD = ORDER_DATE;

(ORDER_DATE is converted from 05/27/93 to 93/05/27.)

A field whose format specifies one set of date components can also be assigned to a field that specifies a different set of date components.

To convert from MDY to Y:

ORDER_YEAR/Y = ORDER_DATE;

(If ORDER_DATE has the format MDY and the value May 27, 1993, then ORDER_YEAR = 93.)

To convert from Y to MDY:

ORDER_DATE/MDY = ORDER_YEAR;

(If ORDER_YEAR has the format Y and the value 1993, then ORDER_DATE will have the value January 1, 1993.)

If there are no values for a portion of the date, FOCUS assigns them the values shown in the chart below.

<u>Portion missing</u>	<u>Portion supplied</u>
day (e.g., from YM to YMD)	first day of the month
month (e.g., from Y to YM)	the month 1 (January)
year from short to long (e.g., from YMD to YYMD)	the year 19xx

Note that these supplied portions are different for the user-written subroutine CHGDAT, which we will discuss next.

● **Converting Components of Dates in Alphanumeric/Integer Format**

The user-written subroutine CHGDAT rearranges the year, month, and day portions of dates and converts between long and short formats.

CHGDAT (informat,outformat,date,outfield)

"Informat" may contain D, M, Y, or YY. "Outformat" may contain D, M, Y, YY, T (3-letter month), or X (full month name). If using a field for "informat" or "outformat", it must be alphanumeric. "Date" must be alphanumeric. "Outfield" can also be the format of the output value in single quotes ('A9').

Example in a table request:

```
DEFINE FILE TEST
ALPHA_HIRE/A6MDY = HIRE_DATE;
END
TABLE FILE TEST
PRINT HIRE_DATE AND COMPUTE
  HIRE2/A9 = CHGDAT ('MDY', 'DMTY', ALPHA_HIRE, 'A9');
BY LAST_NAME BY FIRST_NAME
END
```

If HIRE_DATE has the format MDY, and the value 05/27/93, then HIRE2 will have the value 27 May 93.

If there are no values for a portion of the date, FOCUS assigns them the values shown in the chart below.

Portion missing

day (e.g., from YM to YMD)
month (e.g., from Y to YM)
short year (e.g., from MD to YMD)
long year (e.g., from MD to YYMD)
year from short to long (e.g., from YMD to YYMD)

Portion supplied

last day of the month
the month 12 (December)
the year 99
the year 1999
the year 19xx

Example in Dialogue Manager:

```
-SET &INDATE = 930527;
-SET &DATE1 = &INDATE || 'A';
-SET &DATE2 = CHGDAT('YMD', 'MDY', &DATE1, 'A17');
-TYPE INDATE IS &INDATE , DATE1 IS &DATE1 , DATE2 IS &DATE2

>>INDATE IS 930527 , DATE1 IS 930527A, DATE2 IS 052793
```

Example in Modify:

```
COMPUTE DATE1/A6YMD = '920530';
  DATE2/A17 = CHGDAT ('YMD', 'MDY', DATE1, 'A17');

DATE1 is 930527
DATE2 is 052793
```


3. Arithmetic Operations

• Arithmetic Operations with Internal Date Format

It is easy to perform arithmetic operations on dates that are in FOCUS internal date format. The contents of fields with date formats are stored as integer numbers which represent the difference, in the format you specify (days, years, quarters, etc.), between a base date (December 31, 1900) and the date you supply. Since the contents of the fields are numeric offsets, you can use them in calculations, regardless of the display format you chose (YMD, MDY, etc.).

Master File Description:

```
FILE=TEST, SUFFIX=FIX, $
FIELDNAME = SHIPDATE      ,MDY , $
FIELDNAME = ORDERDATE     ,DMY , $
FIELDNAME = SALESQTR      ,YQ  , $
```

Table Request:

```
DEFINE FILE TEST
IN_PROCESS/I5 = SHIPDATE - ORDERDATE;
END
```

You can subtract two dates using a simple arithmetic expression to determine the elapsed time from ordering to shipping. It gives the correct result, even though the displays of the two fields are different. The date fields must have the same components, but they can be in any order. For example, you could subtract MtrDY from MDY, but you could not subtract YQ from MDY (like SHIPDATE - SALESQTR).

DAYS/I5 = SHIPDATE - '1 JAN 1993'; computes the elapsed number of days between the two dates, regardless of their display format.

LASTWEEK/MDY = '27 MAY 93' - 7; OR

LASTWEEK/MDY = TODAYS_DATE - 7; subtracts one week from today's date. The 7 is interpreted to mean 'days' because the smallest component of the MDY format is days.

SHIPMONTH/MONTH = SHIPDATE; OR

SHIPMONTH/M = SHIPDATE; If SHIPDATE has the value November 24, 1993, then SHIPMONTH is 11.

SHIPMONTH/MONTH = SHIPMONTH + 3; If the current value of SHIPMONTH is 11, then SHIPMONTH + 3 has the value 2. (The range of values is 1 to 12.)

• **Arithmetic Operations with Alphanumeric/Integer Date Format**

The following date functions enable you to calculate the duration between dates if the field format is either alphanumeric or integer.

YMD(begin,end) calculates the duration between two dates stored in the form year-month-day

MDY(begin,end) calculates the duration between two dates stored in the form month-day-year

DMY(begin,end) calculates the duration between two dates stored in the form day-month-year

```
DAYS/I4 = MDY(042093,052793)
DAYS/I4 = DMY(200493,270593)
```

The subtraction is the ending date minus the beginning date. The resulting value is the number of days between the beginning and ending dates, including leap years.

The following user-written subroutines add or subtract a number of days or months from a given date. These subroutines can go beyond the 20th century. "Outfield" can also be the format of the output value in single quotes.

AYMD (indate,days,outfield) adds or subtracts number of days from a given date. "Indate" must be YMD.

```
DEFINE FILE TEST
HIRE_DATE/I6YMD = 930527;
END
TABLE FILE TEST
PRINT HIRE_DATE AND COMPUTE
      AFTER35DAY/I6YMD = AYMD (HIRE_DATE,35,AFTER35DAY);
BY LAST_NAME BY FIRST_NAME
END
```

AYM (indate,months,outfield) adds or subtracts number of months from a given date. "Indate" must be YM. (To convert from an I6YMD or I8YYMD date to the YM format, simply divide by 100.)

```

DEFINE FILE TEST
HIRE_DATE/I6YMD = 930527;
HIRE_MONTH/I4YM = HIRE_DATE / 100;
END
TABLE FILE TEST
PRINT HIRE_DATE HIRE_MONTH AND COMPUTE
      AFTER6MO/I4YM = AYM (HIRE_MONTH, 6, AFTER6MO);
END

```

Output:

HIRE_DATE	HIRE_MONTH	AFTER6MO
93/05/27	93/05	93/11

YM (fromdate,todate,outfield) returns the number of months between two dates. Dates must be in YM. (If they are I6YMD or I8YYMD, simply divide them by 100 to convert to YM format.) "Outfield" can also be the format of the output value in single quotes.

```

DEFINE FILE TEST
HIRE_DATE/I6YMD = 920527;
HIRE_MONTH/I4YM = HIRE_DATE / 100;
TODAYS_DATE/I6YMD = 930527;
TODAYS_MONTH/I4YM = TODAYS_DATE / 100;
END
TABLE FILE TEST
PRINT HIRE_DATE HIRE_MONTH TODAYS_DATE TODAYS_MONTH AND
      COMPUTE MOS_HIRED/I3 = YM (HIRE_MONTH, TODAYS_MONTH, 'I3');
END

```

Output:

HIRE_DATE	HIRE_MONTH	TODAYS_DATE	TODAYS_MONTH	MOS_HIRED
92/05/27	92/05	93/05/27	93/05	12

4. Getting Today's Date and Time

- Using System Variables

&TOD gives the time of day in the format HH.MM.SS
&DATE gives the current date in the format MM/DD/YY
&MDY gives the current date in the format MMDDYY
&DMY gives the current date in the format DDMMYY
&YMD gives the current date in the format YYMMDD

Example in a Table Request:

```
DEFINE FILE TEST
DATE1/A8 = '&DATE';
DATE2/A6 = '&MDY';
DATE3/A6 = '&DMY';
DATE4/A6 = '&YMD';
TIME1/A8 = '&TOD';
END
TABLE FILE TEST
HEADING
"REPORT AS OF <DATE1 AT <TIME1"
"DATE2 = <DATE2"
"DATE3 = <DATE3"
"DATE4 = <DATE4"
PRINT FIRST_NAME
BY LAST_NAME
END
```

Output heading:

```
REPORT AS OF 05/27/93 AT 14.20.51
DATE2 = 052793
DATE3 = 270593
DATE4 = 930527
```

Example in Dialogue Manager:

```
-SET &DATE1 = &DATE;  
-SET &TIME1 = &TOD;  
-TYPE THE DATE IS &DATE1 AND THE TIME IS &TIME1  
  
>>THE DATE IS 05/27/93 AND THE TIME IS 14.20.51
```

The system variables are easy to use, but they have some limitations. Compiled MODIFY requests, for example, cannot use system variables and must use subroutines. Also, note that the system variables are made current only when you enter FOCUS or execute a MODIFY request. Therefore, if you want the exact time or date, use one of the user-written subroutines discussed below. For example, if you run applications late at night, you may prefer to use the TODAY subroutine to get the correct date after midnight.

• Using User-Written Subroutines

TODAY (outfield) displays the current date in an A8 format, month-day-year, with embedded slashes. Unlike the system variables &DATE, &MDY, &DMY, and &YMD, the TODAY subroutine always returns the current date. "Outfield" can also be the format of the output value in single quotes.

HHMMSS (outfield) displays the current time in an A8 format with embedded periods separating the hours, minutes, and seconds. Unlike the system variable &TOD, the HHMMSS subroutine always returns the current time. "Outfield" can also be the format of the output value in single quotes.

Example in a table request:

```
DEFINE FILE TEST
DATE1/A8 WITH EMP_ID = TODAY (DATE1);
TIME1/A8 WITH EMP_ID = HHMMSS ('A8');
END
TABLE FILE TEST
HEADING
"SALARY REPORT AS OF <DATE1 AT <TIME1"
PRINT CURR_SAL
BY LAST_NAME BY FIRST_NAME
END
```

Output heading:

```
REPORT AS OF 05/27/93 AT 14.20.51
```

Note: Since slashes are imbedded in the output from TODAY, FOCUS cannot process it as a date unless you use EDIT to remove the slashes. For example:

```
DEFINE FILE TEST
DATE1/A8 WITH EMP_ID = TODAY (DATE1);
DATE_EDITED/A6MDY WITH EMP_ID = EDIT (DATE1, '99$99$99');
DATE3/MDY WITH EMP_ID = DATE_EDITED
END
```

Example in Dialogue Manager:

```
-SET &DATE1 = TODAY ('A8');
-SET &TIME1 = HHMMSS ('A8');
-TYPE DATE IS &DATE1 AND TIME IS &TIME1

>>DATE IS 05/27/93 AND TIME IS 12.16.47
```

In Dialogue Manager, you must specify the format ('A8'). If you use (&TIME) or ('&TIME'), Dialogue Manager will prompt you for the variable.

5. Calendar Systems

• Converting from Gregorian to Julian

The user-written subroutine JULDAT(indate,outfield) converts dates from year-month-day format to Julian year-day format. Dates in the Julian format are 5-digit numbers. The first two digits represent the year, and the last three represent the number of the day counting from January 1. For example, January 1, 1987 in Julian format is 87001, and December 31, 1987 is 87365. "Outfield" can also be the format of the output value in single quotes.

```
TABLE FILE TEST
PRINT HIRE_DATE AND COMPUTE
      JULIAN_DATE/IS = JULDAT (HIRE_DATE, JULIAN_DATE);
BY LAST_NAME BY FIRST_NAME
END
```

(If HIRE_DATE has the format YMD and the value May 27, 1993, then JULIAN_DATE = 93147.)

• Converting from Julian to Gregorian

The user-written subroutine GREGDT(datej,dateg) converts dates in Julian format to year-month-day format. "Outfield" can also be the format of the output value in single quotes.

```
-PROMPT &JULDATE. ENTER THE JULIAN DATE.
-SET &YMDDATE = GREGDT (&JULDATE, 'I6');
-SET &YMDDATE = EDIT (&YMDDATE, '99/99/99');
-TYPE
-TYPE THE YEAR-MONTH-DAY EQUIVALENT OF &JULDATE IS &YMDDATE.

>>THE YEAR-MONTH-DAY EQUIVALENT OF 93147 IS 93/05/27
```

6. Number of Days this Century

• Converting from Dates to Number of Days this century

The DA subroutines convert dates into the number of days elapsed this century (since January 1, 1900, which is day 1). Don't confuse this with the base date used by the internal date format, which is December 31, 1900. There are six subroutines to accept dates in different formats. Outfield can also be the format of the output value in single quotes.

DAxxx (indate,outfield)
YMD, MDY, DMY, YDM, MYD, DYM

```
DEFINE FILE TEST
BASE_DATE/I6YMD = 0;
DATE1/I6YMD = 930527;
END
TABLE FILE TEST
PRINT BASE_DATE DATE1 AND
      COMPUTE ELAPSED_DAYS/I8 = DAYMD (DATE1,'I8') - DAYMD (BASE_DATE,'I8');
END
```

Output:

BASE_DATE	DATE1	ELAPSED_DAYS
-----	93/05/27	34115

• Converting from Number of Days this century to Dates

DT subroutines convert the number of days elapsed this century (since January 1, 1900, which is day 1) into dates. There are six subroutines to accept dates in different formats. "Outfield" can also be the format of the output value in single quotes.

DTxxx (indate,outfield)
YMD, MDY, DMY, YDM, MYD, DYM

```
-PROMPT &DANUM. ENTER THE NUMBER OF DAYS BETWEEN 1 AND 36,524.
-SET &DATECON = DTYMD (&DAYNUM,'I6');
-SET &DATECON = EDIT (&DATECON,'99/99/99');
-TYPE
-TYPE THE DATE WHICH CORRESPONDS TO DAY &DAYNUM IS &DATECON

>>THE DATE WHICH CORRESPONDS TO DAY 34115 IS 93/05/27
```

Note: When entering small numbers, be aware that the EDIT statement shown above will not give you the correct date format. For example, if you input 25, then &DATECON = 125 (January 25, 1900) and the EDIT statement produces 12/5 / rather than 00/01/25. Likewise, day 1000 produces the result 20/92/7 rather than 02/09/27.

7. Day of the Week

DOWK(indate,outfield) provides the day of the week (3-letter abbreviation, format A4). The subroutine is not limited to the 20th century; it works both forward into the 21st century and backward into the 19th century. "Indate" must be in year-month-day format. If the date specifies a 2-digit year, the subroutine assumes the 20th century. "Outfield" can also be the format of the output value in single quotes.

DOWKL(indate,outfield) provides the full name of the day of the week (format A12).

```
DEFINE FILE TEST
HIRE_DATE/I6YMD = 930527;
END
TABLE FILE TEST
PRINT HIRE_DATE AND COMPUTE
  DOW_S/A4 = DOWK (HIRE_DATE, DOW_S);
  DOW_L/A12 = DOWKL (HIRE_DATE, 'A12');
BY LAST_NAME BY FIRST_NAME
END
```

Output:

```
HIRE_DATE  DOW_S  DOW_L
-----
 93/05/27  THU    THURSDAY
```

8. Moving Into the 21st Century

If we compute the statement 'OCT 15 2002' - 'SEP 15 1998'; we get a response of 1461 days. In the example below, we assign the value of these two literals to date fields.

```
DEFINE FILE TEST
DATE1/MDY = 'OCT 15 2002';
DATE2/MDY = 'OCT 15 1998';
END
TABLE FILE TEST
PRINT ID AND COMPUTE RESULT1/I9 = DATE1 - DATE2;
END
```

In this example, whether you define DATE1 and DATE2 as MDY or MDYY you get the correct result of 1461 days, because the internal date format has the value of the number of elapsed days since the base date, regardless of whether it extends into the 21st century. (It can extend, in fact, several thousand years.) However, you need to be careful how you use the MDY format for dates after the turn of the century. The following two expressions do not give the same value:

```
DATE1/MDY = 'OCT 15 2002';
DATE1/MDY = '10152002';
```

The first is evaluated by FOCUS as October 15, 2002. However, because of the MDY format, FOCUS ignored the first two digits of the year in the second expression and evaluated it incorrectly as October 15, 1902. Consider the following example:

```

DEFINE FILE TEST
DATE1/MDY = 'OCT 15 2002';
DATE2/MDY = '10152002';
BASEDATE/MDY = 0;
END
TABLE FILE TEST
PRINT BASEDATE DATE1 DATE2 AND COMPUTE DIF1/I9 = BASEDATE - DATE1;
                                         DIF2/I9 = BASEDATE - DATE2;
END

```

Output:

BASEDATE	DATE1	DATE2	DIF1	DIF2
-----	-----	-----	-----	-----
	10/15/02	10/15/02	-37178	-653

The moral of the story: Always use a date format with a four-digit year. Not only does it display the whole year (10/15/2002) to remove all doubt, but you are also more sure it will be evaluated correctly.

9. Date Displays

You can temporarily redefine the format of a date directly in the table request as shown in the following example:

```

DEFINE FILE TEST
DATE1/I8MDYY = 05271993;
END
TABLE FILE TEST
PRINT
DATE1/MDY           AS 'MDY'
DATE1/MDYY          AS 'MDYY'
DATE1/YMD           AS 'YMD'
DATE1/YYMD          AS 'YYMD'
DATE1/MTDY          AS 'MTDY'
DATE1/M-D-Y         AS 'M-D-Y'
END

```

Output:

MDY	MDYY	YMD	YYMD	MTDY	M-D-Y
---	----	---	----	----	----
05/27/93	05/27/1993	93/05/27	1993/05/27	MAY 27, 93	05-27-93

Below is a list of the various date display formats:

Format	Display
MDY	06/17/93
MDYY	06/17/1993
YMD	93/06/17
YMYD	1993/06/17
MTDYY	JUN 17, 1993
MTRDYY	JUNE 17, 1993
McDYY	Jun 17, 1993
MtrDYY or McRDYY	June 17, 1993
MTRDYYW	JUNE 17 1993, THU
MTRDYYWR	JUNE 17 1993, THURSDAY
MtrDYYw or McRDYYw	June 17 1993, Thu
McRDYYwR or MtrDYYwR or McRDYYwR	June 17 1993, Thursday
wRMcRDYY or wRMtrDYY or wRMTRDYY	Thursday, June 17 1993
DMTY	17 JUN, 93
DMTRY	17 JUNE, 93
DMtrY or DMcRY	17 June, 93
M-D-Y	06-17-93
M-D-YY	06-17-1993
M.D.Y.	06.17.93
M.D.YY.	06.17.1993
M D Y	06 17 93
M D YY	06 17 1993
MBDBY	06 17 93
MBDBYY	06 17 1993
D	17
W or w	4
WT	THU
WTR	THURSDAY
Wt	Thu
Wtr or WcR	Thursday
M	06
MT	JUN
MTR	JUNE
Mt	Jun
Mtr or Mtr	June
Y	93
YY	1993
Q	Q2
YQ	93 Q2
Y-Q	93-Q2
JUL	93/168

The FOCEXEC which follows uses the Master File Description shown below:

```

FILE=PROJECT, SUFFIX=FOC
SEGNAME=PROJECT, SEGTYPE=S1
FIELDNAME=ID, FORMAT=A5, FIELDTYPE=I, TITLE='PROJ, ID', $
FIELDNAME=TITLE, FORMAT=A20, TITLE='PROJECT TITLE', $
FIELDNAME=APPROVED_SCH, FORMAT=MDYY, TITLE='APPROVAL, SCHEDULED', $
FIELDNAME=APPROVED, FORMAT=MDYY, TITLE='APPROVAL, ACTUAL', $
FIELDNAME=BEGIN_SCH, FORMAT=MDYY, TITLE='BEGIN, SCHEDULED', $
FIELDNAME=BEGIN, FORMAT=MDYY, TITLE='BEGIN, ACTUAL', $
FIELDNAME=COMPLETE_SCH, FORMAT=MDYY, TITLE='COMPLETE, SCHEDULED', $
FIELDNAME=COMPLETE, FORMAT=MDYY, TITLE='COMPLETE, ACTUAL', $
FIELDNAME=DEPARTMENT, FORMAT=A2, TITLE='DEPT', $

```

The FOCEXEC extracts information from the file PROJECT and prepares three different reports utilizing different date display formats.

```

--*
--* DATE_REPORT BY LARRY RICHMAN MAY 27, 1993
--*
SET PRINT=OFFLINE
SET PAGE-NUM=OFF
SET LINES=43
SET PANEL=180
SET WIDTH=180
--*
DEFINE FILE PROJECT
DATEB/I6MDY = &MDY;
TODAY_MDY/MDY = DATEB;
TODAY_A6/A6DMTY = 'SDMY';
APP_SCH_1/ASMDY = APPROVED_SCH;
APP_SCH_A9/A9 = CHGDAT('MDY', 'DMTY', APP_SCH_1, 'A9');
APP_1 /ASMDY = APPROVED ;
APP_A9 /A9 = CHGDAT('MDY', 'DMTY', APP_1 , 'A9');
BEG_SCH_1/ASMDY = BEGIN_SCH ;
BEG_SCH_A9/A9 = CHGDAT('MDY', 'DMTY', BEG_SCH_1, 'A9');
BEG_1 /ASMDY = BEGIN ;
BEG_A9 /A9 = CHGDAT('MDY', 'DMTY', BEG_1 , 'A9');
COM_SCH_1/ASMDY = COMPLETE_SCH;
COM_SCH_A9/A9 = CHGDAT('MDY', 'DMTY', COM_SCH_1, 'A9');
COM_1 /ASMDY = COMPLETE ;
COM_A9 /A9 = CHGDAT('MDY', 'DMTY', COM_1 , 'A9');
LATE_DATE/I3 WITH ID =
IF COMPLETE EQ 0 AND COMPLETE_SCH LT TODAY_MDY THEN 1 ELSE 0;
LATE_FLAG/A2 WITH ID =
IF COMPLETE EQ 0 AND COMPLETE_SCH LT TODAY_MDY THEN '***' ELSE ' ';
COUNT_PROJ/I3 WITH ID = 1;
DEPT_NAME/A10 = DECODE DEPARTMENT (OF 'OPERATIONS'
PE 'PERSONNEL ');
END

```

```

TABLE FILE PROJECT
HEADING
" PROJECTS FOR OPERATIONS AS OF <TODAY_MDY PAGE <TABPAGE NO </1"
" TO: ROBERT REED (NEW YORK)"
" FROM: LARRY RICHMAN (PUBLICATIONS COORDINATION)"
" "
PRINT LATE_FLAG AS ''
ID IN 3
TITLE
APPROVED_SCH/MDY AS 'APPROVAL,SCHEDULED'
APPROVED/MDY AS 'APPROVAL,ACTUAL'
BEGIN_SCH/MDY AS 'BEGIN,SCHEDULED'
BEGIN/MDY AS 'BEGIN,ACTUAL'
COMPLETE_SCH/MDY AS 'COMPLETE,SCHEDULED'
COMPLETE/MDY AS 'COMPLETE,ACTUAL'
BY DEPT_NAME NOPRINT PAGE-BREAK
SUBFOOT
" "
" TOTAL PROJECTS = <TOT.COUNT_PROJ"
SUBFOOT
"***NUMBER OF LATE PROJECTS = <TOT.LATE_DATE"
WHEN LATE_DATE GT 0
SUBFOOT
" ALL JOBS ARE ON SCHEDULE"
WHEN LATE_DATE EQ 0
IF DEPARTMENT EQ 'OP'
END

```

```

TABLE FILE PROJECT
HEADING
" PROJECTS FOR PERSONNEL AS OF <TODAY_A6 PAGE <TABPAGENO </1"
" TO: WOLFGANG PAUL (FRANKFURT, GERMANY)"
" FROM: LARRY RICHMAN (PUBLICATIONS COORDINATION)"
" "
PRINT LATE_FLAG AS ''
      ID          IN 3
      TITLE
      APP_SCH_A9 AS 'APPROVAL,SCHEDULED'
      APP_A9      AS 'APPROVAL,ACTUAL'
      BEG_SCH_A9 AS 'BEGIN,SCHEDULED'
      BEG_A9      AS 'BEGIN,ACTUAL'
      COM_SCH_A9 AS 'COMPLETE,SCHEDULED'
      COM_A9      AS 'COMPLETE,ACTUAL'
BY DEPT_NAME NOPRINT PAGE-BREAK
SUBFOOT
" "
" TOTAL PROJECTS = <TOT.COUNT_PROJ"
SUBFOOT
"***NUMBER OF LATE PROJECTS = <TOT.LATE_DATE"
WHEN LATE_DATE GT 0
SUBFOOT
" ALL JOBS ARE ON SCHEDULE"
WHEN LATE_DATE EQ 0
IF DEPARTMENT EQ 'PE'
END

```

```

TABLE FILE PROJECT
HEADING
" PROYECTOS DE OPERACIONES Y PERSONAL <TODAY_A6 PAGINA
<+0<TABPAGENO </1"
" A: NOEL PINO (CARACAS, VENEZUELA)"
" DE: LARRY RICHMAN (COORDINACION DE PUBLICACIONES)"
" "
PRINT LATE_FLAG AS ''
      ID          AS 'NUM ' IN 3
      TITLE       AS 'TITULO'
      APP_SCH_A9 AS 'APROBACION,PROYECTADA'
      APP_A9      AS 'APROBACION,TERMINADA'
      BEG_SCH_A9 AS 'COMIENZO,PROYECTADA'
      BEG_A9      AS 'COMIENZO,TERMINADA'
      COM_SCH_A9 AS 'DISPONIBLE,PROYECTADA'
      COM_A9      AS 'DISPONIBLE,TERMINADA'
BY TABPAGENO NOPRINT
SUBFOOT
" "
" TOTAL DE PROYECTOS = <TOT.COUNT_PROJ"
SUBFOOT
"***HAY <TOT.LATE DATE PROYECTOS MARCADOS CON ASTERISCOS"
" INDICA QUE LAS FECHAS DE IMPRESION SE VENCIERON"
WHEN LATE_DATE GT 0
SUBFOOT
" TODOS LOS PROYECTOS ESTAN AL DIA"
WHEN LATE_DATE EQ 0
IF DEPARTMENT EQ 'OP' OR 'PE'
END
FIN

```

Output:

PROJECTS FOR OPERATIONS AS OF 05/27/93 PAGE 1

TO: ROBERT REED (NEW YORK)
FROM: LARRY RICHMAN (PUBLICATIONS COORDINATION)

PROJ ID	PROJECT TITLE	APPROVAL SCHEDULED	APPROVAL ACTUAL	BEGIN SCHEDULED	BEGIN ACTUAL	COMPLETE SCHEDULED	COMPLETE ACTUAL
33011	PROCEDURE MANUAL	04/01/93	04/02/93	04/02/93	04/03/93	05/10/93	05/18/93
**33022	POLICY MANUAL	02/25/93	03/01/93	02/25/93	03/02/93	03/30/93	
33123	MANAGER MANUAL	01/15/93	01/21/93	01/16/93	01/22/93	02/15/93	05/01/93
**33153	ADMINISTRATOR MANUAL	01/15/93	01/20/93	05/01/93	01/15/93	05/26/93	
33423	SUPERVISOR MANUAL	12/15/92	12/15/92	12/16/92	12/17/93	05/30/93	

TOTAL PROJECTS = 5
**NUMBER OF LATE PROJECTS = 2

PROJECTS FOR PERSONNEL AS OF 27 MAY 93 PAGE 1

TO: WOLFGANG PAUL (FRANKFURT, GERMANY)
FROM: LARRY RICHMAN (PUBLICATIONS COORDINATION)

PROJ ID	PROJECT TITLE	APPROVAL SCHEDULED	APPROVAL ACTUAL	BEGIN SCHEDULED	BEGIN ACTUAL	COMPLETE SCHEDULED	COMPLETE ACTUAL
33033	PERSONNEL MANUAL	02 FEB 93	02 FEB 93	03 FEB 93	03 FEB 93	25 FEB 93	10 MAY 93
33127	EMPLOYEE MANUAL	14 JAN 93	14 JAN 93	15 JAN 93	15 JAN 93	30 MAY 93	

TOTAL PROJECTS = 2
ALL JOBS ARE ON SCHEDULE

A: NOEL PINO (CARACAS, VENEZUELA)
 DE: LARRY RICHMAN (COORDINACION DE PUBLICACIONES)

NUM	TITULO	APROBACION PROYECTADA	APROBACION TERMINADA	COMIENZO PROYECTADA	COMIENZO TERMINADA	DISPONIBLE PROYECTADA	DISPONIBLE TERMINADA
33011	PROCEDURE MANUAL	01 APR 93	02 APR 93	02 APR 93	03 APR 93	10 MAY 93	18 MAY 93
**33022	POLICY MANUAL	25 FEB 93	01 MAR 93	25 FEB 93	02 MAR 93	30 MAR 93	
33033	PERSONNEL MANUAL	02 FEB 93	02 FEB 93	03 FEB 93	03 FEB 93	25 FEB 93	10 MAY 93
33123	MANAGER MANUAL	15 JAN 93	21 JAN 93	16 JAN 93	22 JAN 93	15 FEB 93	01 MAY 93
33127	EMPLOYEE MANUAL	14 JAN 93	14 JAN 93	15 JAN 93	15 JAN 93	30 MAY 93	
**33153	ADMINISTRATOR MANUAL	15 JAN 93	20 JAN 93	01 MAY 93	15 JAN 93	26 MAY 93	
33423	SUPERVISOR MANUAL	15 DEC 92	15 DEC 92	16 DEC 92	17 DEC 93	30 MAY 93	

TOTAL DE PROYECTOS = 7

**HAY 2 PROYECTOS MARCADOS CON ASTERISCOS
 INDICA QUE LAS FECHAS DE IMPRESION SE VENCIERON

To display month names in Spanish, perform additional date conversions as illustrated below:

```

DEFINE FILE PROJECT ADD
MON_S1/A3 = EDIT(APP_SCH_A9,'$$$999$$$');
MON_S/A3=DECODE MON_S1 (JAN ENE
                        FEB FEB
                        MAR MAR
                        APR ABR
                        MAY MAY
                        JUN JUN
                        JUL JUL
                        AUG AGO
                        SEP SEP
                        OCT OCT
                        NOV NOV
                        DEC DIC);
APP_SCH_SP/A9 = EDIT(APP_SCH_A9,'999$$$$$') |
MON_S |
EDIT(APP_SCH_A9,'$$$$$999');
END
    
```