There are multiple **Date Functions** by which you can reformat input dates.

* **TOJUL=Yaa** – converts to a Julian date without a separator (for example, P’2009007′).
* **TOJUL=Yaa(s)** – converts to a Julian date with a separator (for example, C’325-2008′).
* **TOGREG=Yaa** – converts to a Gregorian date without separators (for example, Z’091121′).
* **TOGREG=Yaa(s)** – converts to a gregorian date with separators (for example, C’2009.09.21′).
* **WEEKDAY=CHAR3** – converts to a 3 character day of the week (for example, C’WED’ for Wednesday).
* **WEEKDAY=CHAR9** – converts to a 9 character day of the week (for example, C’THURSDAY ‘ for Thursday).
* **WEEKDAY=DIGIT1** – converts to a 1 digit indicator for the day of the week (for example, C’2′ for Monday).

## **Current date as character string**

**DATE1, &DATE1, DATE1(c), &DATE1(c), DATE2, &DATE2, DATE2(c), &DATE2(c), DATE3, &DATE3, DATE3(c), &DATE3(c), DATE4, &DATE4, DATE5 or &DATE5** can be used to generate a character string for the current date of the run.

**Note**: You can precede each of the operands in the table with an & with identical results. When a field is shorter than the character string it’s compared to, DFSORT truncates the string on the right. You can take advantage of this to compare a field to only part of the DATE4 timestamp when appropriate. For example:

**INCLUDE COND=(1,13,CH,GT,DATE4)**

The above condition will compare the field in positions 1-13 to the truncated DATE4 constant C’yyyy-mm-dd-hh’.

| **Character Strings for Current DateFormat of Operand** | **Format of Constant** | **Example of Constant** |
| --- | --- | --- |
| DATE1 | C’yyyymmdd’ | C’20050621′ |
| DATE1(c) | C’yyyycmmcdd’ | C’2005/06/21′ |
| DATE2 | C’yyyymm’ | C’200506′ |
| DATE2(c) | C’yyyycmm’ | C’2005/06′ |
| DATE3 | C’yyyyddd’ | C’2005172′ |
| DATE3(c) | C’yyyycddd’ | C’2005/172′ |
| DATE4 | C’yyyy-mm-dd-hh.mm.ss’ | C’2005-06-21-16.52.45′ |
| DATE4 | C’yyyy-mm-dd-hh.mm.ss’ | C’2005-06-21-16.52.45′ |
| DATE5 | C’yyyy-mm-dd-hh.mm.ss.nnnnnn’ | C’2005-06-21-16.52.45.582013′ |

## **Future or Past date as character string**

**DATE1+d, &DATE1+d, DATE1(c)+d, &DATE1(c)+d, DATE2+m, &DATE2+m, DATE2(c)+m, &DATE2(c)+m, DATE3+d, &DATE3+d, DATE3(c)+d or &DATE3(c)+d** can be used to generate a character string for a future date relative to the current date of the run. d is days in the future and m is months in the future. d and m can be 0 to 9999.

**DATE1-d, &DATE1-d, DATE1(c)-d, &DATE1(c)-d, DATE2-m, &DATE2-m, DATE2(c)-m, &DATE2(c)-m, DATE3-d, &DATE3-d, DATE3(c)-d or &DATE3(c)-d** can be used to generate a character string for a past date relative to the current date of the run. d is days in the future and m is months in the future. d and m can be 0 to 9999.d is days in the future and m is months in the future. d and m can be 0 to 9999.

## **Current date as decimal number**

**DATE1P, &DATE1P, DATE2P, &DATE2P, DATE3P, or &DATE3P** can be used to generate a decimal number for the current date of the run.

**DATE1P+d, &DATE1P+d, DATE2P+m, &DATE2P+m, DATE3P+d, or &DATE3P+d** can be used to generate a decimal number for a future date relative to the current date of the run. d is days in the future and m is months in the future. d and m can be 0 to 9999.

**DATE1P-d, &DATE1P-d, DATE2P-m, &DATE2P-m, DATE3P-d, or &DATE3P-d** can be used to generate a decimal number for a past date relative to the current date of the run. d is days in the past and m is months in the past. d and m can be 0 to 9999.

## **Date Functions** : Date Formating

### **YYYYMMDD to YYYYDDD format**

The following job converts a ‘yyyymmdd’ date to a ‘yyyyddd’ date:

**//S1 EXEC PGM=SORT**

**//SYSOUT DD SYSOUT=\***

**//SORTIN DD \***

**20090520**

**20100106**

**20100921**

**20081217**

**//SORTOUT DD SYSOUT=\***

**//SYSIN DD \***

**OPTION COPY**

**INREC BUILD=(1,8,Y4T,TOJUL=Y4T)**

**/\***

**OUTPUT:**

2009140

2010006

2010264

2008352

### **YYMMDD to YYYY-DDD format**

**INPUT:**

ABC 090520

DEF 100106

GHI 100921

JKL 081217

**SYSIN:**

**OPTION Y2PAST=1990**

**SORT FIELDS=(5,6,Y2T,A)**

**OUTREC OVERLAY=(5:5,6,Y2T,TOJUL=Y4T(-))**

**OUTPUT:**

JKL 2008-352

ABC 2009-140

DEF 2010-006

GHI 2010-264

### **DDDYYYY to MM-DD-YYYY format**

**OPTION COPY**

**OUTFIL OVERLAY=(1:1,4,Y4X,TOGREG=Y4W(-))**

### **YYYY-MM-DD to MMDDYY format**

**SORT FIELDS=(1,10,CH,A)**

**OUTREC IFTHEN=(WHEN=INIT,BUILD=(1,10,UFF,TO=ZD,LENGTH=8)),**

**IFTHEN=(WHEN=INIT,BUILD=(1,8,Y4T,TOGREG=Y2Y))**

### **Converts MM/DD/YYYY to YYYYDDD format**

**OPTION COPY**

**OUTREC IFTHEN=(WHEN=INIT,**

**BUILD=(1,10,UFF,TO=ZD,LENGTH=8,9:11,3)),**

**IFTHEN=(WHEN=INIT,BUILD=(1,8,Y4W,TOJUL=Y4T,9,3))**

## **Date Functions: Validate Input Date**

You can use TOGREG or TOJUL functions to identify invalid input dates. Dates with values outside of the valid range (for example, a month not between 01-12) will be shown as asterisks making them easy to identify.

**Example: If you had the following input records with ‘yyyymmdd’ dates:**

**INPUT**:

Betten 20091021

Vezinaw 20091101

Casad 00000000

Boenig 20091325

Kolusu 20090931

Yaeger 20090731

You could use below control statements to display an additional column with asterisks for any invalid dates

**SYSIN:**

OPTION COPY

OUTREC OVERLAY=(30:16,8,Y4T,TOGREG=Y4T)

**OUTPUT**:

BETTEN 20091021 20091021

VEZINAW 20091101 20091101

CASAD 00000000 00000000

BOENIG 20091325 \*\*\*\*\*\*\*\*

KOLUSU 20090931 \*\*\*\*\*\*\*\*

YAEGER 20090731 20090731

If you wanted to display only the records with invalid dates, you could use these control statements:

**SYSIN:**

OPTION COPY OUTREC OVERLAY=30:16,8,Y4T,TOGREG=Y2T) OUTFIL INCLUDE=30,1,CH,EQ,C'\*'),BUILD=(1,25)

**OUTPUT:** BOENIG 20091325 KOLUSU 20090931

## **Date Functions:** **Calculate days between two date fields**

**Here is an example on how to calculate the number of days between two dates.**

**INPUT:**

20101215 20101105

20110218 20100913

20110127 20110305

**//S1 EXEC PGM=SORT**

**//SYSOUT DD SYSOUT=\***

**//SORTIN DD DSN=... input file**

**//SORTOUT DD DSN=... output file**

**//SYSIN DD \***

**OPTION COPY**

**INREC OVERLAY=(20:1,8,Y4T,DATEDIFF,10,8,Y4T)**

/\*

**OUTPUT:**

20101215 20101105 +0000040

20110218 20100913 +0000158

20110127 20110305 -0000037

## **Add/subtract days, months, and years from a date fields**

You can use the following date arithmetic functions:

* **ADDDAYS, ADDMONS and ADDYEARS** can be used to add days, months or years to a date field.
* **SUBDAYS, SUBMONS and SUBYEARS** can be used to subtract days, months or years from a date field.
* **DATEDIFF** can be used to calculate the number of days between two date fields.
* **NEXTDday** can be used to calculate the next specified day of the week for a date field (where day can be SUN, MON, TUE, WED, THU, FRI or SAT). **NEXTDFRI** can be used to decide the next Friday for a C’ccyyddd’ date as a C’ccyy.ddd’ date:
* **PREVDday** can be used to calculate the previous specified day of the week for a date field (where day can be SUN, MON, TUE, WED, THU, FRI or SAT).
* **LASTDAYW, LASTDAYM, LASTDAYQ and LASTDAYY** can be used to calculate the last day of the week, month, quarter or year for a date field.

### **Date Functions** **Example**

**//STEP0100 EXEC PGM=SORT**

**//SYSOUT DD SYSOUT=\***

**//SORTIN DD \***

**20101215**

**20110110**

**20110225**

**//SORTOUT DD SYSOUT=\***

**//SYSIN DD \***

**SORT FIELDS=COPY**

**INREC OVERLAY=(15:1,8,Y4T,ADDDAYS,+15,TOGREG=Y4T(-),**

**30:1,8,Y4T,SUBDAYS,+23,TOGREG=Y4T(-))**

**/\***

This job adds 15 days to a ‘yyyymmdd’ date in input positions 1-8 and converts the result to a ‘yyyy-mm-dd’ date in output positions 15-24. Subtracts 23 days from a ‘yyyymmdd’ date in input positions 1-8 and converts the result to a ‘yyyy-mm-dd’ date in output positions 30-39.

**OUTPUT:**

20101215 2010-12-30 2010-11-22

20110110 2011-01-25 2010-12-18

20110225 2011-03-12 2011-02-02

Use the following to calculate the next Friday for a C’ccyyddd’ date as a C’ccyy.ddd’ date: 3,7,Y4T,NEXTDFRI,TOJUL=Y4T(.)

Use the following to calculate the previous Wednesday for a P’yyddd’ date as a C’ccyymmdd’ date: 51,3,Y2U,PREVDWED,TOGREG=Y4T

Use the following to calculate the last day of the month for a C’mmddccyy’ date as a C’mmddccyy’ date: 28,8,Y4W,LASTDAYM,TOGREG=Y4W

This next job subtracts 3 months from a ‘yyddd’ date in input positions 1-5 and converts the result to a ‘dddyyyy’ date in output positions 11-17.

**Sort by date, and calculate a specific day after and before a date, and the last day of the quarter for a date. The input date is in the form C’mmddyy’ and the output dates will be in the form ‘ddd-yyyy’.**

**INPUT:**

010105

120699

021610

999999

092810

031500

000000

032505

110210

**SYSIN:**

**OPTION Y2PAST=1990**

**SORT FIELDS=(1,6,Y2W,A)**

**OUTFIL REMOVECC, HEADER1=(1:'Input',15:'NEXTDFRI',25:'PREVDSUN',35:'LASTDAYQ'),**

**BUILD=(1:1,6,Y2W,TOJUL=Y4W(-),**

**15:1,6,Y2W,NEXTDFRI,TOJUL=Y4W(-),**

**25:1,6,Y2W,PREVDSUN,TOJUL=Y4W(-),**

**35:1,6,Y2W,LASTDAYQ,TOJUL=Y4W(-))**

**OUTPUT**:

INPUT NEXTDFRI PREVDSUN LASTDAYQ

000-0000 000-0000 000-0000 000-0000

340-1999 075-2000 001-2005 084-2005

047-2010 271-2010 306-2010 999-9999

344-1999 339-1999 365-1999 077-2000

072-2000 091-2000 007-2005 361-2004

090-2005 091-2005 079-2005 090-2005

050-2010 045-2010 090-2010 274-2010

269-2010 273-2010 309-2010 304-2010

365-2010 999-9999 999-9999 999-9999

Note that the ‘000000’ and ‘999999’ input values are treated as special indicators for output.

### **Date Functions Example**

**//STEP0200 EXEC PGM=SORT**

**//SYSOUT DD SYSOUT=\***

**//SORTIN DD \***

**10036**

**11017**

**11122**

**//SORTOUT DD SYSOUT=\***

**//SYSIN DD \***

**OPTION Y2PAST=1980**

**SORT FIELDS=(1,5,Y2T,D)**

**OUTREC BUILD=(1,5,5X,1,5,Y2T,SUBMONS,+3,TOJUL=Y4W)**

/\*

**OUTPUT:**

11122 0332011

11017 2902010

10036 3092009

## **Extract corresponding weekdays from dates**

**WEEKDAY function** can be used to extract the day of the week from various types of dates. Three different output formats for the date are supported as follows:

* **DIGIT1 – returns 1** digit for the weekday corresponding to the date (‘1’ for Sunday through ‘7’ for Saturday).
* **CHAR3 – returns 3** characters for the weekday corresponding to the date (‘SUN’ for Sunday through ‘SAT’ for Saturday).
* **CHAR9 – returns 9** characters for the weekday corresponding to the date (‘SUNDAY ‘ for Sunday through ‘SATURDAY ‘ for Saturday).

For example, if you use this job:

**//S1 EXEC PGM=SORT**

**//SYSOUT DD SYSOUT=\***

**//SORTIN DD \***

**07132009**

**07152009**

**07172009**

**//SORTOUT DD SYSOUT=\***

**//SYSIN DD \***

**SORT FIELDS=COPY**

**INREC OVERLAY=(15:1,8,Y4W,WEEKDAY=DIGIT1,X**

**1,8,Y4W,WEEKDAY=CHAR3,X,**

**1,8,Y4W,WEEKDAY=CHAR9)**

**/\***

**OUTPUT:**

07132009 2 MON MONDAY

07152009 4 WED WEDNESDAY

07172009 6 FRI FRIDAY

**If you wanted just the CHAR9 result, but with initial capitals, you could use these DFSORT control statements:**

**SYSIN:**

**SORT FIELDS=COPY**

**INREC OVERLAY=(15:1,8,Y4W,WEEKDAY=CHAR9,**

**16:16,8,TRAN=UTOL)**

**OUTPUT:**

07132009 Monday

07152009 Wednesday

07172009 Friday