

## **DFSORT's Year 2000 Features (The New Generation)**

October, 1998

Frank L. Yaeger

DFSORT Team  
IBM Storage Systems Division  
San Jose, California  
VNET: YAEGER at SJFEVMX  
IBMMAIL: USIB257Z at IBMMAIL  
Internet: fyaeger@vnet.ibm.com

### **DFSORT/MVS Web Site**

Come visit DFSORT/MVS on the World Wide Web at URL:

<http://www.ibm.com/storage/dfsort/>



---

## Abstract

DFSORT has **Year 2000 features** you can use to sort, merge, compare, and transform a wide variety of dates with **two-digit years** according to a specified sliding or fixed century window.

DFSORT's **new generation** of Year 2000 features handle **full dates** like yyq, yymm, yyddd, yymmdd, qyy, mmyy, dddy and mmddy and their **special indicators** like zeros and nines. You no longer have to split dates into year and non-year pieces, and you don't need E61 exits to handle special indicators.

This paper contains a detailed discussion of DFSORT's Year 2000 features and examples of the control statements needed to order, compare, and transform all kinds of character, zoned decimal and packed decimal date fields.

This paper also explains how to use DFSORT's new Year 2000 features with **COBOL**, either automatically with **COBOL Millennium Language Extensions (MLE)** or explicitly without MLE.



---

# Contents

DFSORT's Year 2000 Features	1
Overview	1
Century Window	2
Full Date Formats	3
Year Formats	5
Sort and Merge on Dates	6
Compare Dates	10
Transform Dates	13
Report on Dates	19
Use COBOL MLE	21
Use DFSORT's Year 2000 Features with COBOL	21
SORT Examples (Order)	25
INCLUDE Examples (Compare)	32
OUTFIL OUTREC Examples (Transform)	35



---

## DFSORT's Year 2000 Features

DFSORT has **Year 2000 features** you can use to sort, merge, compare, and transform a wide variety of dates with **two-digit years**, according to a specified sliding or fixed century window.

These Year 2000 features are available in:

- DFSORT Release 14 with PTF **UQ22534**
- DFSORT Release 13 with PTFs **UN90139**, **UQ05520** and **UQ22533**.

DFSORT's **new generation** of Year 2000 features handle **full dates** like yyq, yymm, yyddd, yymmdd, qyy, mmyy, dddy and mmddy and their **special indicators** like zeros and nines. You no longer have to split dates into year and non-year pieces, and you don't need E61 exits to handle special indicators.

**Note:** If you haven't installed PTF UQ22534 or UQ22533, you will not be able to use the full date formats or INCLUDE/OMIT support described in this paper, but you will be able to use the year formats.

This paper contains a detailed discussion of DFSORT's Year 2000 features and examples of the control statements needed to order, compare, and transform all kinds of character, zoned decimal and packed decimal date fields.

This paper also explains how to use DFSORT's new Year 2000 features with **COBOL**, either automatically with **COBOL Millennium Language Extensions (MLE)** or explicitly without MLE.

For information on the many features of DFSORT mentioned in this paper such as OUTFIL and ICETOOL, see one or more of the following:

- *DFSORT Application Programming Guide (SC33-4035)*
- The DFSORT home page at URL:  
<http://www.ibm.com/storage/dfsor/>
- The **SORTBE14** "Beyond Sorting" paper available via anonymous FTP to:  
<ftp.software.ibm.com/storage/dfsor/mvs/>

## Overview

The widespread use of two-digits to represent years can result in significant data processing problems. For example, the normal ordering of 00 before 99 is incorrect when 00 represents 2000 and 99 represents 1999. DFSORT's Year 2000 features provide tools that can help you correctly process a wide variety of dates for this century and the coming ones.

DFSORT provides powerful Year 2000 features that allow you to sort, merge, compare, and transform character, zoned decimal and packed decimal dates with two-digit years according to a specified sliding or fixed century window.

You can handle two-digit year date fields, and their special indicators like zeros and nines, in the following ways:

- Set the appropriate **century window** for your applications and use it to interpret the years (yy) correctly when you sort, merge, compare or transform two-digit year dates. For example, set a century window of 1915-2014 or 1950-2049.

- **Order** character, zoned decimal or packed decimal two-digit year dates with the SORT and MERGE statements. For example, order 980622 (representing June 6th, 1998) before 000622 (representing June 6th, 2000) in ascending sequence, or order 000622 before 980622 in descending sequence.

DFSORT's **full date formats** (Y2T, Y2U, Y2V, Y2W, Y2X and Y2Y) make it easy to sort or merge any type of yyx...x or x...xyy date (for example, yyq, yymm, yyddd, yymmdd, qyy, mmyy, dddy or mmdyy). DFSORT's **year formats** (Y2C, Y2Z, Y2S, Y2P, Y2D and Y2B) are available if you need to sort or merge special dates (for example, Z'ddmmyy' or C'yy/ddd') or year fields (yy).

- **Select** records by **comparing** character, zoned decimal or packed decimal two-digit year date fields to date constants or other date fields with the INCLUDE and OMIT statements or OUTFIL operands. For example, select records with a date field between January 1st, 1998 and December 31st, 2003.

DFSORT's full date formats make it easy to compare any type of yyx...x or x...xyy date field to a date constant or another date field. DFSORT's year formats are available if you need to compare years.

- **Transform** character, zoned decimal or packed decimal two-digit year dates to character four-digit year dates with or without separators, or transform character, zoned decimal or packed decimal two-digit year dates to packed decimal four-digit year dates, with DFSORT's OUTFIL OUTREC operand. For example, transform P'99015' to C'1999015', C'1999/015' or P'1999015'.

DFSORT's full date formats make it easy to transform any type of yyx...x or x...xyy date. DFSORT's year formats are available if you need to transform special dates or year fields.

In addition, you can use DFSORT's Year 2000 features with COBOL, either automatically with **COBOL MLE** or explicitly without MLE.

These DFSORT enhancements allow you to continue to use two-digit year dates for sorting, merging and comparing, and help you to change from using two-digit year dates to using four-digit year dates as appropriate.

## Century Window

A new **installation** and **run-time** option allows you to specify the **century window** you want DFSORT to use for interpreting two-digit years. A century window spans a 100 year period, such as 1950-2049, and is used to control how the two-digit years 00 to 99 are interpreted. For example, 56 will be interpreted as 1956 if you set the century window to 1950-2049 or as 2056 if you set the century window to 1990-2089.

DFSORT allows you to set either a **fixed** century window or a **sliding** century window as follows:

- **Y2PAST=f** specifies a **fixed** century window starting at f, where f can be any value between 1000 and 3000. For example, Y2PAST=1985 starts the century window at 1985, which always sets the century window to 1985-2084.

You might want to use a fixed century window for two-digit year data that you started collecting in some particular year (for example, 1985 in this case) or to examine historic or future two-digit year data (for example, 1200-1299 or 2500-2599).

- **Y2PAST=s** specifies a **sliding** century window starting s years before the current year, where s can be any value between 0 and 100. For example, if the current year is 1998, Y2PAST=80 starts the century window at 1998 - 80 = 1918, which sets the century window to 1918-2017. When 1999 rolls around, Y2PAST=80 starts the century window at 1999 - 80 = 1919, which sets the century window to 1919-2018. You can see that Y2PAST=s causes the century window to automatically slide year by year.

You might want to use a sliding century window for two-digit year data you collect over a rolling 30 year period.

Thus, the Y2PAST option allows you to control how DFSORT interprets the two-digit years 00 to 99 for various sets of data. You can use the Y2PAST **installation** option to set a default site-wide fixed or sliding century window. The shipped IBM installation default is Y2PAST=80 which sets a sliding window starting 80 years before the current year. You can use the Y2PAST **run-time** option to set specific fixed or sliding century windows for different applications.

As an example of setting the century window, both Y2PAST=1915, and Y2PAST=83 used in 1998, give a century window of 1915 to 2014, which results in the following interpretation of two-digit years by DFSORT.

yy	Interpreted as
--	-----
00	2000
14	2014
15	1915
61	1961
62	1962
99	1999

## Full Date Formats

DFSORT provides six **full date formats** that make it easy to sort, merge, compare and transform character, zoned decimal and packed decimal yyx...x and x...xyy two-digit year dates. The **Y2T, Y2U, Y2V, Y2W, Y2X and Y2Y** formats (f) can be used with various lengths (m) to handle dates like yyq, yymm, yyddd, yymmdd, qyy, mmyy, dddyy and yymmdd and "special indicators" commonly used with these dates.

Table 1 shows each full date format (m,f) along with its applicable dates and special indicators. y indicates a year digit (0-9), x indicates a non-year digit (0-9), and s indicates a sign (0-F). The following notation is used for the special indicators:

- Character: C'0...0' (CH zeros), Z'0...0' (ZD zeros), C'9...9' (CH nines) and Z'9...9' (ZD nines)
- Binary: X'00...00' (BI zeros), X'40...40' (blanks) and X'FF...FF' (BI ones)
- Packed: P'0...0' (PD zeros) and P'9...9' (PD nines)

Table 1. Full Date Formats, Dates and Special Indicators

<b>m,f</b>	<b>Applicable Dates</b>	<b>Special Indicators</b>
3,Y2T	C'yyx', Z'yyx'	Character, Binary
4,Y2T	C'yyxx', Z'yyxx'	Character, Binary
5,Y2T	C'yyxxx', Z'yyxxx'	Character, Binary
6,Y2T	C'yyxxxx', Z'yyxxxx'	Character, Binary
2,Y2U	P'yyx' (X'yyxs')	Packed
3,Y2V	P'yyxx' (X'0yyxxs')	Packed
3,Y2U	P'yyxxx' (X'yyxxxs')	Packed
4,Y2V	P'yyxxxx' (X'0yyxxxxs')	Packed
3,Y2W	C'xyy', Z'xyy'	Character, Binary
4,Y2W	C'xxyy', Z'xxyy'	Character, Binary
5,Y2W	C'xxxxy', Z'xxxxy'	Character, Binary
6,Y2W	C'xxxxxy', Z'xxxxxy'	Character, Binary
2,Y2X	P'xyy' (X'xyys')	Packed
3,Y2Y	P'xxyy' (X'0xxyys')	Packed
3,Y2X	P'xxxxy' (X'xxxxyys')	Packed
4,Y2Y	P'xxxxxy' (X'0xxxxxyys')	Packed

DFSORT will interpret "real" two-digit years (yy) according to the century window established by the Y2PAST option in effect. DFSORT will not use the century window for special indicators.

For maximum processing efficiency, DFSORT assumes that:

- Y2T and Y2W dates consist of C'd...d', Z'd...d', X'00...00', X'40...40' or X'FF...FF' values where d is 0-9.
- Y2U, Y2V, Y2X and Y2Y dates consist of P'd...d' values where d is 0-9.

Any other values in these dates may produce unexpected results.

Table 2 is a handy reference chart for the full date formats. It shows each type of date with an example of that type, and the full date format (m,f) to use for a corresponding character (CH), zoned decimal (ZD) or packed decimal (PD) date.

Table 2. Quick Reference for Full Date Formats

Type of Date	Example of Type	CH/ZD (m,f)	PD (m,f)
yyx	yyq	3,Y2T	2,Y2U
yyxx	yymm	4,Y2T	3,Y2V
yyxxx	yyddd	5,Y2T	3,Y2U
yyxxxx	yymmdd	6,Y2T	4,Y2V
xyy	qyy	3,Y2W	2,Y2X
xyyy	mmyy	4,Y2W	3,Y2Y
xxxyy	dddy	5,Y2W	3,Y2X
xxxxyy	mmddy	6,Y2W	4,Y2Y

Using these tables, you can pick the right full date format for any yyx...x or x...xyy two-digit year date. For example:

- Say you have a C'yyddd' date field starting at position 18. You've been using 18,5,CH for its position, length and format. But now you need to use windowing for this two-digit year date, so you need a windowing format rather than CH. You can see from Table 1 or Table 2 that you should use 18,5,Y2T for this date (i.e. substitute Y2T for CH). You can see from Table 1 that CH zeros, CH nines, BI zeros, blanks and BI ones will be handled as special indicators with this full date format.

If you wanted to sort this C'yyddd' date field in ascending order, you could use the following SORT statement:

```
SORT FIELDS=(18,5,Y2T,A)
```

- If you have a P'mmddy' date starting at position 25, you can see from Table 1 or Table 2 that you should use 25,4,Y2Y for its position, length and format. You can see from Table 1 that PD zeros and PD nines will be handled as special indicators with this full date format.

If you wanted to sort this P'mmddy' date field in descending order, you could use the following SORT statement:

```
SORT FIELDS=(25,4,Y2Y,D)
```

**Note:** To use full date formats, x...x must be in the correct order for sorting or merging. For example, you can use full date formats for yymmdd and mmddy dates (a yyxxx type) because month (mm) followed by day (dd) will sort or merge correctly. However, you cannot use full date formats for yydmm or ddmyy dates because day followed by month will not sort or merge correctly. You would need to use a year format for a yydmm or ddmyy date as explained in "Year Formats."

"SORT Examples (Order)" on page 25, "INCLUDE Examples (Compare)" on page 32 and "OUTFIL OUTREC Examples (Transform)" on page 35 show you how to use the full date formats (and year formats) for many different dates.

## Year Formats

In most cases, you can use DFSORT's full date formats to handle your two-digit year dates. However, DFSORT also provides six **year formats**. You can use the **Y2C, Y2Z, Y2S, Y2P, Y2D and Y2B** formats to handle:

- Character, zoned decimal, binary and packed decimal year fields (yy).
- Special dates that the full date formats cannot handle. For example, you can use the yyxxx full date formats for a yymmdd date, but not for a yydmm date. The reason is that yymmdd has month (mm)

followed by day (dd), which will sort or merge correctly, but yyddmm has day followed by month, which will not sort correctly. Likewise, you can use the xxxxyy full date formats for an mmdyy date, but not for a ddmmy date. To sort or merge a yyddmm or ddmmy date, you must specify the year (yy) using Y2C followed by the month (mm) using BI and then the day (dd) using BI.

Table 3 shows each year format (m,f) along with its applicable years and special indicators. y indicates a year digit (0-9). i indicates an ignored digit (0-F). hh indicates a hexadecimal value (00-FF) representing a decimal value (000-255) whose last two digits are to be used as yy. (Typically, X'00' to X'63' would be used to represent 00 to 99 and values beyond X'63' (99) would not be used.)

Table 3. Year Formats, Years and Special Indicators

<b>m,f</b>	<b>Applicable Years</b>	<b>Special Indicators</b>
2,Y2C	C'yy', Z'yy' (X'iyiy')	None
2,Y2Z	C'yy', Z'yy' (X'iyiy')	None
2,Y2S	C'yy', Z'yy' (X'iyiy')	X'0000', X'4040', X'FFFF'
2,Y2P	P'yy' (X'iyyi')	None
1,Y2D	X'yy'	None
1,Y2B	X'hh' (binary)	None

DFSORT will interpret "real" two-digit years (yy) according to the century window established by the Y2PAST option in effect. DFSORT will not use the century window for special indicators.

For packed decimal dates, you may also need DFSORT's **PD0** format to handle parts of PD fields such as ddmm. PD0 can be used for any two- to eight-byte packed decimal field you want to treat as:

i d d d . . . d i

where **i** (0-F) is ignored and **d** represents a decimal digit (0-9).

As an example of using a year format and PD0, if you wanted to sort a P'yyddmm' (X'0yyddmms') date field starting in position 1 in ascending order, you could use the following SORT statement:

```
SORT FIELDS=(1,2,Y2P,A,3,2,PD0,A,2,2,BI,A)
```

"SORT Examples (Order)" on page 25 and "OUTFIL OUTREC Examples (Transform)" on page 35 show you how to use the year formats and PD0 format (as well as the full date formats) for many different dates.

"INCLUDE Examples (Compare)" on page 32 shows you how to use the year formats for many different year fields (and the full date formats for many different dates).

## Sort and Merge on Dates

You can use DFSORT's **full date formats** (Y2T, Y2U, Y2V, Y2W, Y2X and Y2V) to correctly sort and merge any character, zoned decimal or packed decimal date of the form yyx...x or x...xyy, according to the century window you specify. The full date formats will order dates and special indicators as follows when used in SORT or MERGE statements:

- Y2T and Y2W (for character and zoned decimal dates)
  - Ascending: BI zeros, blanks, CH/ZD zeros, lower century dates (e.g. 19yy), upper century dates (e.g. 20yy), CH/ZD nines, BI ones.

- Descending: BI ones, CH/ZD nines, upper century dates, lower century dates, CH/ZD zeros, blanks, BI zeros.
- Y2U, Y2V, Y2X and Y2Y (packed decimal dates)
  - Ascending: PD zeros, lower century dates, upper century dates, PD nines.
  - Descending: PD nines, upper century dates, lower century dates, PD zeros.

You can use DFSORT's **year formats** (Y2C, Y2Z, Y2S, Y2P, Y2D and Y2B) to correctly sort and merge year fields and special dates, according to the century window you specify.

Y2S will order years and special indicators as follows when used in SORT or MERGE statements:

- Ascending: X'0000', X'4040', lower century years (e.g. 19yy), upper century years (e.g. 20yy), X'FFFF'.
- Descending: X'FFFF', upper century years, lower century years, X'4040', X'0000'.

All other year formats do not handle special indicators. They will order the years as follows when used in SORT or MERGE statements:

- Ascending: lower century years, upper century years.
- Descending: upper century years, lower century years.

We'll go through some explanation and examples here to help you understand how to set up the SORT and MERGE control statements for some typical date fields. But to save you the trouble of reinventing the wheel, a fairly complete list of examples of SORT statements for date fields is provided in "SORT Examples (Order)" on page 25.

## Explanation

You always want to order dates by **four-digit year**, then **month** (if present) and then **day** (if present). So if you have a date in the form mmddyy, you would order it as yyyyymmdd, using the century window to interpret yy as yyyy. Likewise, if you have a date in the form yyddd, you would order it as yyyyddd. The full date formats take care of this for you automatically, but if you use year formats you have to take care of it yourself.

When you use full date formats or year formats, you need to decide whether to use the site's default century window (CW) or set a different century window for the application using the Y2PAST run-time option.

Although a date field can start anywhere in a record and can be only one of a number of fields you are ordering on, for the sake of simplicity we will assume in the examples below that you are only sorting on one date field which starts in position 1.

When you sort or merge character (CH), zoned decimal (ZD) and packed decimal (PD) dates, you can view them as being represented as follows in hexadecimal:

```
CH/ZD:  FdFd...sd
PD:     dd...ds
```

where **d** is a decimal digit (0-9) and **s** is a sign (0-F). For dates, the sign is meaningless and can be ignored.

There are many types of CH, ZD and PD dates. Here are some examples: C'mmddyy', Z'yyq', C'yyddd', Z'mmyy', C'yy-mm-dd', P'mmddyy' (X'0mmddyys'), P'yyddd' (X'yyddd's'), P'dddyy' (X'ddyys'), P'yymm' (X'0yymms').

To sort or merge CH, ZD or PD dates, use DFSORT's full date formats whenever possible. Table 1 shows each full date format (m,f) along with its applicable dates and special indicators. To sort or merge year

fields, use the year formats shown in Table 2. If you want to sort or merge a date that's not shown in Table 1, you'll have to use year and non-year formats. This can be tricky because you have to split up the date and it's not always obvious how to do that, especially for PD dates where the year, month and day can be split across bytes. The full date formats take care of all the complexities automatically, but you have to do it yourself if you use the year formats. In either case, you can use "SORT Examples (Order)" on page 25 to help you set up the SORT statement for your dates; the list of dates with examples is quite extensive.

## Examples

Here's an example of control statements to sort a **C'mmddy'** date field in ascending order. Table 1 shows that 6,Y2W can handle a CH xxxxyy date field, so that's the full date format we need. If you look up C'mmddy' in "SORT Examples (Order)" on page 25, you can find its index (09) and go directly to its SORT statement examples.

```
* Set 1962-2061 as the fixed century window
OPTION Y2PAST=1962
* Sort C'mmddy' as C'yyyymmdd'
SORT FIELDS=(1,6,Y2W,A) * Sort C'mmddy' using CW
```

Here are some results for the specified century window of 1962-2061. Note that the special indicator (blanks) are sorted in the correct position relative to the real dates.

SORTIN input (date field is C'mmddy')

```
090196 * represents 1996/09/01 *
111461 * represents 2061/11/14 *
100305 * represents 2005/10/03 *
      * special indicator - blanks *
031862 * represents 1962/03/18 *
100205 * represents 2005/10/02 *
100302 * represents 2002/10/03 *
```

SORTOUT output (date field is C'mmddy')

```
      * special indicator - blanks *
031862 * represents 1962/03/18 *
090196 * represents 1996/09/01 *
100302 * represents 2002/10/03 *
100205 * represents 2005/10/02 *
100305 * represents 2005/10/03 *
111461 * represents 2061/11/14 *
```

Here's an example of a control statement to sort a **Z'yyddd'** date field in **descending** order. Table 1 shows that 5,Y2T can handle a ZD yyxxx date field, so that's the full date format we need. If you look up Z'yyddd' in "SORT Examples (Order)" on page 25, you can find its index (03) and go directly to its SORT statement examples.

```
* Use site's default century window of 1916-2015
* Sort Z'yyddd' as Z'yyyddd'
SORT FIELDS=(1,5,Y2T,D) * Sort Z'yyddd' using CW
```

Here are some results for the site's default century window of 1916-2015. Note that the special indicators (zeros and nines) are sorted in the correct positions relative to the real dates. Remember that the records were sorted in **descending** order.

SORTIN input (date field is Z'yyddd')

```
00048 * represents 2000/048 *
62125 * represents 1962/125 *
00000 * special indicator - zeros *
16228 * represents 1916/228 *
15022 * represents 2015/022 *
99999 * special indicator - nines *
99095 * represents 1999/095 *
62003 * represents 1962/003 *
```

SORTOUT output (date field is Z'yyddd')

```
99999 * special indicator - nines *
15022 * represents 2015/022 *
00048 * represents 2000/048 *
99095 * represents 1999/095 *
62125 * represents 1962/125 *
62003 * represents 1962/003 *
16228 * represents 1916/228 *
00000 * special indicator - zeros *
```

Here's an example of a control statement to sort a **P'mmddy' (X'0mddyys') date field** in ascending order. Table 1 shows that 4,Y2Y can handle a PD xxxxyy date field, so that's the full date format we need. If you look up P'mmddy' in "SORT Examples (Order)" on page 25, you can find its index (10) and go directly to its SORT statement examples.

```
* Use site's default century window of 1916-2015
* Sort P'mmddy' (X'0mddyys') as P'yyyymmdd'
SORT FIELDS=(1,4,Y2Y,A) * Sort P'mmddy' using CW
```

Here are some results for the site's default century window of 1916-2015. Note that the special indicators (zeros and nines) are sorted in the correct positions relative to the real dates.

SORTIN input (date field is P'mmddy' - shown as X'0mddyys')

```
0042800C * represents 2000/04/28 *
0110199C * represents 1999/11/01 *
0032562C * represents 1962/03/25 *
0999999C * special indicator - nines *
0042816C * represents 1916/04/28 *
0000000C * special indicator - zeros *
0062215C * represents 2015/06/22 *
0031862C * represents 1962/03/18 *
```

SORTOUT output (date field is P'mmddy' - shown as X'0mddyys')

```
0000000C * special indicator - zeros *
0042816C * represents 1916/04/28 *
0031862C * represents 1962/03/18 *
0032562C * represents 1962/03/25 *
0110199C * represents 1999/11/01 *
0042800C * represents 2000/04/28 *
0062215C * represents 2015/06/22 *
0999999C * special indicator - nines *
```

Here's an example of control statements to sort a **C'mm/dd/yy' date field** in ascending order (the "/" could be replaced by any separator character). Table 1 shows that the full date formats cannot handle a CH xx/xx/yy date field, so we'll need to use the Y2C year format for the year (yy) and the BI format for the month (mm)

and day (dd). Remember that year formats do not handle special indicators except for those handled by Y2S. If you look up C'mm/dd/yy' in "SORT Examples (Order)" on page 25, you can find its index (30) and go directly to its SORT statement example.

```
* Set 1962 to 2061 as the fixed century window
OPTION Y2PAST=1962
* Sort C'mm/dd/yy' as C'yyyymmdd'
SORT FIELDS=(7,2,Y2C,A, * Sort C'yy' using CW
              1,2,BI,A, * Sort C'mm'
              1,4,BI,A) * Sort C'dd'
```

Here are some results for the specified century window of 1962-2061.

SORTIN input (date field is C'mm/dd/yy')

```
09/01/98 * represents 1998/09/01 *
11/14/61 * represents 2061/11/14 *
10/03/05 * represents 2005/10/03 *
03/18/62 * represents 1962/03/18 *
10/03/02 * represents 2002/10/03 *
```

SORTOUT output (date field is C'mm/dd/yy')

```
03/18/62 * represents 1962/03/18 *
09/01/98 * represents 1998/09/01 *
10/03/02 * represents 2002/10/03 *
10/03/05 * represents 2005/10/03 *
11/14/61 * represents 2061/11/14 *
```

## Compare Dates

You can use DFSORT's Y2 formats and Y date constant in INCLUDE and OMIT statements and in the INCLUDE and OMIT operands of OUTFIL statements.

DFSORT's **full date formats** (Y2T, Y2U, Y2V, Y2W, Y2X and Y2V) and **Y date constant** allow you to select records for output by comparing date fields to date constants or to other date fields. Any character, zoned decimal or packed decimal date of the form yyx...x or x...xyy can be compared to a date constant or to another date field with the same number of non-year digits, according to the century window you specify.

DFSORT's **year formats** (Y2C, Y2Z, Y2S, Y2P, Y2D and Y2B) and **Y year constant** allow you to select records for output by comparing year fields to year constants or to other year fields. Any character, zoned decimal, packed decimal or binary year can be compared to a year constant or another year field, according to the century window you specify.

The ordering of dates and special indicators used for comparisons is the same as the ascending orders shown in "Sort and Merge on Dates" on page 6. The following comparison operators can be used with date field comparisons: EQ, NE, GT, GE, LT and LE.

We'll go through some explanation and examples here to help you understand how to set up the INCLUDE and OMIT control statements or OUTFIL operands for some typical date fields. But to save you the trouble of reinventing the wheel, a fairly complete list of examples of INCLUDE statements for date fields is provided in "INCLUDE Examples (Compare)" on page 32.

## Explanation

You always want to compare dates on both sides of the comparison operator by **four-digit year**, then **month** (if present) and then **day** (if present). So if you're comparing an mmddyy date field to a date constant, you want both the date and the constant in the form yyymmdd, using the century window to interpret yy as yyyy for both. The full date format and Y date constant take care of this for you automatically.

When you use full date formats or year formats, you need to decide whether to use the site's default century window (CW) or set a different century window for the application using the Y2PAST run-time option.

When you compare character (CH), zoned decimal (ZD) and packed decimal (PD) dates, you can view them as being represented as follows in hexadecimal:

```
CH/ZD:  FdFd...sd
PD:     dd...ds
```

where **d** is a decimal digit (0-9) and **s** is a sign (0-F). For dates, the sign is meaningless and can be ignored.

There are many types of CH, ZD and PD dates. Here are some examples: C'mmddyy', Z'yyq', C'yyddd', Z'mmyy', C'yymmdd', P'mmddyy' (X'0mmddys'), P'yyddd' (X'yydds'), P'dddy' (X'ddys'), P'yymm' (X'0yymms').

A date field can be compared to a Y date constant with the same number of non-year (x) digits (e.g. a C'yyxxx' field or P'xxxxy' field can be compared to a Y'yyxxx' constant). A date field can also be compared to any other date field with the same number of non-year (x) digits (e.g. a P'xxxxy' field can be compared to a Z'xxxxy' field or a P'yyxxxx' field). A year field can be compared to a Y year constant or to any other year field (e.g. P'yy' can be compared to a Y'yy' constant or to a C'yy' field).

Table 4 shows the type of field-to-constant and field-to-field comparisons you can use with the full date formats and year formats. The fields shown for any type of date (e.g. yyx and xyy) can be compared to any other fields shown for that type of date or to the Y constant shown for that type of date.

Table 4. Date Fields and Constants for Comparisons

Type of Date	Fields (m,f)	Y Constant
yyx and xyy	3,Y2T   2,Y2U   3,Y2W   2,Y2X	Y'yyx'
yyxx and xxxy	4,Y2T   3,Y2V   4,Y2W   3,Y2Y	Y'yyxx'
yyxxx and xxxyy	5,Y2T   3,Y2U   5,Y2W   3,Y2X	Y'yyxxx'
yyxxxx and xxxxyy	6,Y2T   4,Y2V   6,Y2W   4,Y2Y	Y'yyxxxx'
yy	2,Y2C   2,Y2Z   2,Y2S   2,Y2P   1,Y2D   1,Y2B	Y'yy'

You must use the same number of digits in a Y constant as the type of date; **leading zeros** must be specified (e.g. for Y'yymm', use Y'0001' for January, 2000 and Y'0101 for January, 2001).

You can also use Y constants for special indicators as follows:

- Y'0...0' (CH/ZD/PD zeros) and Y'9...9' (CH/ZD/PD nines) can be used with Y2T, Y2U, Y2V, Y2W, Y2X and Y2Y dates. You must use the same number of digits as the type of date (e.g. Y'000' for yyq or qyy, Y'0000' for yymm or mmyy, etc.).
- Y'LOW' (BI zeros), Y'BLANKS' (blanks) and Y'HIGH' (BI ones) can be used with Y2T, Y2W and Y2S dates.

## Examples

Here's an example of control statements to select records for which a **C'yymm' date field** starting in position 3 is **between January, 1999 and December, 2003**. Table 1 shows that 4,Y2T can handle a CH yymm date field, so that's the full date format we need. Table 4 shows that a yymm date can be compared to a Y'yymm' constant, so that's the Y constant we need. If you look up C'yymm' in "INCLUDE Examples (Compare)" on page 32, you can find its index (05) and go directly to its INCLUDE statement example.

```
* Set 1967-2066 as the fixed century window
OPTION Y2PAST=1967,COPY
* Compare a C'yymm' field to Y'yymm' constants
INCLUDE FORMAT=Y2T, * Specify full date format for all fields
      COND=(3,4,GE,Y'9901',AND, * Include if field ge 1999/01
            3,4,LE,Y'0312') * and le 2003/12
```

If you decided to also select records for which the C'yymm' date is all CH zeros (i.e. C'0000'), you could change your control statements to:

```
* Set 1967-2066 as the fixed century window
OPTION Y2PAST=1967,COPY
* Compare a C'yymm' field to Y'yymm' constants
INCLUDE FORMAT=Y2T, * Specify full date format for all fields
      COND=(3,4,GE,Y'9901',AND, * Include if field ge 1999/01
            3,4,LE,Y'0312',OR, * and le 2003/12, or
            3,4,EQ,Y'0000') * eq zeros indicator
```

Here are some results for the specified century window of 1967-2066. Note that the zeros special indicator is included in SORTOUT, but the nines special indicator is not.

SORTIN input (date field is C'yymm')

```
0401 * represents 2004/01 *
0005 * represents 2000/05 *
9908 * represents 1999/08 *
0312 * represents 2003/12 *
0000 * special indicator - zeros *
9812 * represents 1998/12 *
9999 * special indicator - nines *
9901 * represents 1999/01 *
0511 * represents 2005/11 *
```

SORTOUT output (date field is C'yymm')

```
0005 * represents 2000/05 *
9908 * represents 1999/08 *
0312 * represents 2003/12 *
0000 * special indicator - zeros *
9901 * represents 1999/01 *
```

Here's an example of control statements to select records for which a **P'dddy' date field** starting in position 2 is less than a **Z'yyddd' date field** starting in position 36. Table 1 shows that 3,Y2X can handle a PD dddy date and that 5,Y2T can handle a ZD yyddd date, so those are the full date formats we need. As a double-check, Table 4 shows that a 3,Y2X field can be compared to a 5,Y2T field. If you look up P'dddy' in "INCLUDE Examples (Compare)" on page 32, you can find its index (12) and go directly to its INCLUDE statement example. Likewise, if you look up Z'yyddd', you can find its index (03) and go directly to its INCLUDE statement.

```

* Use site's default century window of 1916-2015
* Sort P' dddy' as P' yyyyddd'
  SORT FIELDS=(2,3,Y2X,A)      * Sort P' dddy'
* Compare P' dddy' as yyyyddd to Z' yyddd' as yyyyddd
OUTFIL FNAMES=OUT1,
  INCLUDE=(2,3,Y2X,LT,36,5,Y2T) * Include if field 1 lt field 2

```

Here are some results for the specified century window of 1916-2015. Note that the special indicators are compared correctly.

```

SORTIN input (field 1 is P' dddy' - shown as X' dddyys',
              field 2 is Z' yyddd')

```

```

10503C <- represents 2003/105      01003 <- represents 2001/003
12395C <- represents 1995/123      95123 <- represents 1995/123
05102C <- represents 2002/051      99999 <- nines indicator
12803C <- represents 2003/128      14062 <- represents 2014/062
00000C <- zeros indicator          99003 <- represents 1999/003
99999C <- nines indicator          92023 <- represents 1992/023
00598C <- represents 1998/005      00010 <- represents 2000/010

```

```

OUT1 output (field 1 is P' dddy' - shown as X' dddyys',
             field 2 is Z' yyddd')

```

```

00000C <- zeros indicator          99003 <- represents 1999/003
00598C <- represents 1998/005      00010 <- represents 2000/010
05102C <- represents 2002/051      99999 <- nines indicator
12803C <- represents 2003/128      14062 <- represents 2014/062

```

## Transform Dates

You can use DFSORT's **full date formats** in the OUTREC operand of the OUTFIL statement, to transform character, zoned decimal and packed decimal dates of the form yx...x or x...xyy, according to the century window you specify:

- Y2T, Y2U, Y2V, Y2W, Y2X and Y2Y can be used to transform character, zoned decimal and packed decimal two-digit year dates to character four-digit year dates, with or without separators (e.g. transform Z'mmddy' to C'mmddy' or C'mm-dd-yyy').
- Y2TP, Y2UP, Y2VP, Y2WP, Y2XP and Y2YP can be used to transform character, zoned decimal and packed decimal two-digit year dates to packed decimal four-digit year dates (e.g. transform P'yyddd' to P'yyyyddd').

Full date formats handle all of the special indicators by expanding them (e.g. if you are transforming P'yymm' to C'yyyy/mm', a P'9999' date will be expanded to C'9999/99').

You can use DFSORT's **year formats** in the OUTREC operand of the OUTFIL statement, to transform year fields and special dates, according to the century window you specify:

- Y2C, Y2Z, Y2S, Y2P, Y2D and Y2B can be used to transform character, zoned decimal, packed decimal and binary two-digit year fields to character four-digit year fields. Y2S handles its special indicators by expanding them (e.g. if you are transforming C'yy' to C'yyyy', a X'FFFF' date will be expanded to X'FFFFFFF'). All other year formats do not handle special indicators.
- Y2PP and Y2DP can be used to transform packed decimal two-digit year fields to packed decimal four-digit year fields.

We'll go through some explanation and examples here to help you understand how to set up the OUTFIL control statements for transforming some typical date fields. But to save you the trouble of reinventing the wheel, a fairly complete list of examples of OUTFIL statements for transforming date fields is provided in "OUTFIL OUTREC Examples (Transform)" on page 35.

## Explanation

You can use the OUTFIL date transformation techniques shown here for a variety of tasks such as to create new data sets with four-digit years or to produce reports containing transformed dates.

When you use full date formats or year formats, you need to decide whether to use the site's default century window (CW) or set a different century window for the application using the Y2PAST run-time option.

When you transform character (CH), zoned decimal (ZD) and packed decimal (PD) dates, you can view them as being represented as follows in hexadecimal:

```
CH: FdFd...Fd
ZD: FdFd...sd
PD: dd...ds
```

**d** is a decimal digit (0-9). **s** is the sign (0-F) for ZD and PD values. **C**, a positive sign, is normally used for **s**.

There are many types of CH, ZD and PD dates. Here are some examples: C'mmddy', Z'yyq', C'yyddd', Z'mmyy', C'yy-mm-dd', P'mmddy' (X'0mmddyys'), P'yyddd' (X'yyddd's'), P'dddy' (X'dddyys'), P'yymm' (X'0yymm's').

To transform CH, ZD or PD dates, use DFSORT's full date formats whenever possible.

- p,m,Y2x transforms a CH, ZD or PD two-digit year date to a CH four-digit year date (e.g. p,3,Y2T transforms Z'yyq' to C'yyyyq'). Table 5 shows the output produced when p,m,Y2x is used.
- p,m,Y2x(c) transforms a CH, ZD or PD two-digit year date to a CH four-digit year date with separators (e.g. p,3,Y2X(-) transforms P'dddy' to C'ddd-yyyy'). **c** can be any character (e.g. "/", "-", ".", etc) **except** a blank. Table 5 shows the output produced when p,m,Y2x(/) is used.
- p,m,Y2xP transforms a CH, ZD or PD two-digit year date to a PD four-digit year date (e.g. p,3,Y2VP transforms P'yymm' to P'yyyyymm'). Table 6 shows the output produced when p,m,Y2xP is used.

If you want to transform a date that cannot be handled by the full date formats, you'll have to use year and non-year formats. This can be tricky because you have to split up the date and it's not always obvious how to do that, especially for PD dates where the year, month and day can be split across bytes. The full date formats take care of all the complexities automatically, but you have to do it yourself if you use the year formats. In either case, you can use "OUTFIL OUTREC Examples (Transform)" on page 35 to help you set up the OUTFIL statement for your dates; the list of dates with examples is quite extensive.

Table 5. p,m,Y2x and p,m,Y2x(c) Output

Type of Date	Fields (m,f)	Output for p,m,Y2x	Output for p,m,Y2x(/)
yyx	3,Y2T   2,Y2U	C'yyyyx'	C'yyyy/x'
yyxx	4,Y2T   3,Y2V	C'yyyyxx'	C'yyyy/xx'
yyxxx	5,Y2T   3,Y2U	C'yyyyxxx'	C'yyyy/xxx'
yyxxxx	6,Y2T   4,Y2V	C'yyyyxxxx'	C'yyyy/xx/xx'
xyy	3,Y2W   2,Y2X	C'xyyyy'	C'x/yyyy'
xxyy	4,Y2W   3,Y2Y	C'xxyyyy'	C'xx/yyyy'
xxxxy	5,Y2W   3,Y2X	C'xxxxyyy'	C'xxx/yyyy'
xxxxyy	6,Y2W   4,Y2Y	C'xxxxyyyy'	C'xx/xx/yyyy'

Table 6. p,m,Y2xP Output

Type of Date	Fields (m,f)	Output
yyx	3,Y2TP   2,Y2UP	P'yyyyx'
yyxx	4,Y2TP   3,Y2VP	P'yyyyxx'
yyxxx	5,Y2TP   3,Y2UP	P'yyyyxxx'
yyxxxx	6,Y2TP   4,Y2VP	P'yyyyxxxx'
xyy	3,Y2WP   2,Y2XP	P'xyyyy'
xxyy	4,Y2WP   3,Y2YP	P'xxyyyy'
xxxxy	5,Y2WP   3,Y2XP	P'xxxxyyy'
xxxxyy	6,Y2WP   4,Y2YP	P'xxxxyyyy'

## Examples

Here's an example of control statements to transform a **Z'mmddy'** date field in SORTIN to a **C'mm/dd/yyyy'** date field in NEW1. Assume that the SORTIN data set has a **Z'mmddy'** date field in position 21 and is fixed format with a logical record length of 200. Since the 6 digit zoned decimal date field will become a 10 character date field, the NEW1 data set will have a logical record length of 204.

Table 1 shows that 6,Y2W can handle a ZD xxxxy date. Table 5 shows that 6,Y2W(/) will produce a C'xx/xx/yyyy' output field, so that's the full date format we need. If you look up **Z'mmddy'** in "OUTFIL OUTREC Examples (Transform)" on page 35, you can find its index (13) and go directly to its OUTFIL statement examples.

\* Use site's default century window of 1916-2015

OPTION COPY

\* Transform Z'mmddy' (6 bytes) to C'mm/dd/yyyy' (10 bytes) in NEW1

OUTFIL FAMES=NEW1,

```

OUTREC=(1,20,      * copy positions 1-20
        21,6,Y2W(/), * Z'mmddy' -> C'mm/dd/yyyy'
        27,174)    * copy the rest of the record

```

Here are some results for the site's default century window of 1916-2015. Note that the special indicator (nines) is just expanded, whereas the real dates are transformed using the century window.

SORTIN input date field in position 21  
(Z'mmddy' - shown as C'mmddy' - note last character)

12049E  
03210C  
99999I  
05088B

NEW1 output date field in position 21 (C'mmddyyyy')

12/04/1995  
03/21/2003  
99/99/9999  
05/08/1982

Here's an example of control statements to sort and transform a **P'yyddd' (X'yyddd's) date field** in SORTIN to a **C'yyyyddd' date field** in CHNOSEP, a **C'yyy.ddd' date field** in CHSEP and a **P'yyyyddd' (X'yyyydddC) date field** in PACK. Assume the P'yyddd' date starts in position 1 and we want to sort it in **descending** order.

Table 1 shows 3,Y2U can handle a PD yyxxx date.

- Table 5 shows that 3,Y2U will produce a C'yyyyxxx' output field, so that's the full date format we need in the OUTFIL statement for CHNOSEP.
- Table 5 shows that 3,Y2U(.) will produce a C'yyy.ddd' output field, so that's the full date format we need in the CHSEP OUTFIL statement for CHSEP.
- Table 6 shows that 3,Y2UP will produce a P'yyyyddd' output field, so that's the full date format we need in the OUTFIL statement for PACK.

If you look up P'yyddd' in "OUTFIL OUTREC Examples (Transform)" on page 35, you can find its index (06) and go directly to its OUTFIL statement examples.

```
* Use century window of 1980-2079
OPTION Y2PAST=1980
* Sort on P'yyddd' in descending order
SORT FIELDS=(1,3,Y2U,D)
* Transform P'yyddd' (3 bytes) to C'yyyyddd' (7 bytes) in CHNOSEP
OUTFIL FNAMES=CHNOSEP,OUTREC=(1,3,Y2U)
* Transform P'yyddd' (3 bytes) to C'yyy.ddd' (8 bytes) in CHSEP
OUTFIL FNAMES=CHSEP,OUTREC=(1,3,Y2U(.))
* Transform P'yyddd' (3 bytes) to P'yyyyddd' (4 bytes) in PACK
OUTFIL FNAMES=PACK,OUTREC=(1,3,Y2UP)
```

Here are some results for the specified century window of 1980-2079.

SORTIN input (date field is P'yyddd' - shown as X'yydds')

80123C  
98202C  
79052C  
00003C  
98105C  
02026C

CHNOSEP output (date field is C'yyyyddd')

2079052  
2002026  
2000003  
1998202  
1998105  
1980123

CHSEP output (date field is C'yyy.ddd')

2079.052  
2002.026  
2000.003  
1998.202  
1998.105  
1980.123

PACK output (date field is P'yyyyddd' - shown as X'yyyydddC')

2079052C  
2002026C  
2000003C  
1998202C  
1998105C  
1980123C

Here's an example of control statements to transform a **C'yymmdd' date field** in SORTIN to a **P'yyyymmdd' (X'0yyyymmddC') date field** in NEW2. Assume the C'yymmdd' date starts in position 1. The 6 digit character field will become a 5-digit packed decimal field. In order to maintain the original position and length of the date field, we'll add one blank after it.

Table 1 shows that 6,Y2T can handle a CH yyxxxx date. Table 6 shows that 6,Y2TP will produce a P'yyyyxxxx' output field, so that's the full date format we need. If you look up C'yymmdd' in "OUTFIL OUTREC Examples (Transform)" on page 35, you can find its index (01) and go directly to its OUTFIL statement examples.

\* Use site's default century window of 1916-2015

OPTION COPY

\* Transform C'yymmdd' (6 bytes) to P'yyyymmdd' (5 bytes) in NEW2

OUTFIL FAMES=NEW2,

OUTREC=(1,6,Y2TP, \* C'yymmdd' -> P'yyyymmdd'

X, \* add one blank

7,40) \* copy the rest of the record

Here are some results for the site's default century window of 1916-2015. Note that the special indicator (zeros) is just expanded, whereas the real dates are transformed using the century window.

SORTIN input (date field is C'yymmdd')

951204 \* represents 1995/12/04 \*  
000101 \* represents 2000/01/01 \*  
000000 \* special indicator - zeros \*  
990628 \* represents 1999/06/28 \*

NEW2 output (date field is P'yyyymmdd' - shown as X'0yyymmddC')

019951204C \* represents 1995/12/04 \*  
020000101C \* represents 2000/01/01 \*  
000000000C \* special indicator - zeros \*  
019990628C \* represents 1999/06/08 \*

Here's an example of control statements to transform a **C'mm.dd.yy' date field** in SORTIN to a **C'yyyy-mm-dd' date field** in SORTOUT. Assume the C'mm.dd.yy' date starts in position 1.

Table 1 shows that the full date formats cannot handle a CH xx/xx/yy date field, so we'll need to use the Y2C format for the year and copy the month (mm) and day (dd). Remember that year formats do not handle special indicators except for those handled by Y2S. If you look up C'mm/dd/yy' (equivalent to C'mm.dd.yy') in "OUTFIL OUTREC Examples (Transform)" on page 35, you can find its index (36) and go directly to its OUTFIL examples. Note that the examples transform C'mm/dd/yy' to C'mm/dd/yyyy' so you'll have to extrapolate a bit to transform C'mm.dd.yy' to C'yyyy-mm-dd'.

```
* Set 1962 to 2061 as the fixed century window
OPTION COPY,Y2PAST=1962
* Transform C'mm/dd/yy' (8 bytes) to C'yyyy-mm-dd' (10 bytes)
* in SORTOUT.
OUTFIL OUTREC=(7,2,Y2C, * change C'yy' to C'yyyy' using CW
                  C'-', * insert dash
                  1,2, * copy C'mm'
                  C'-', * insert dash
                  4,2) * copy C'dd'
```

Here are some results for the specified century window of 1962-2061.

SORTIN input (date field is C'mm.dd.yy')

09.01.96  
11.14.61  
10.03.05  
03.18.62  
10.03.02

SORTOUT output (date field is C'yyyy-mm-dd')

1996-09-01  
2061-11-14  
2005-10-03  
1962-03-18  
2002-10-03

## Report on Dates

Using the techniques explained in “Transform Dates” on page 13, you can use **OUTFIL** to transform dates so that they can be used in reports.

For a given application, you can either:

- use **OUTFIL** to transform the dates and create the report all at once, or
- use **OUTFIL** to transform the dates and then use **ICETOOL** to create the report.

Choosing to create the report with **ICETOOL** vs **ICETOOL** is a matter of the **time** you want to spend on the report and how much **control** you need over the way it looks.

## OUTFIL Reports

Transformed dates can easily be incorporated directly into **OUTFIL reports** by including them in the **OUTREC operand**.

Here’s an example of JCL and control statements to produce an **OUTFIL** report with a P’yymmdd’ date field (starting in position 21) transformed into a C’yyyy-mm-dd’ date field.

```
//REPORT1 EXEC PGM=ICEMAN
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=TRANS.INPUT,DISP=SHR
//RPT1 DD SYSOUT=*
//SYSIN DD *
* Use century window of 1990-2089
OPTION Y2PAST=1990
* Sort on P’yymmdd’
SORT FIELDS=(21,4,Y2V,A)
* Create report with P’yymmdd’ (X’0yymmddC’) transformed
* to C’yyyy-mm-dd’
OUTFIL FNames=RPT1,
  HEADER2=(2:C’ Transaction Report for ’,DATE=(4MD-),2/,
           2:C’ Date ’,17:C’ Item Code’,31:C’ Price ($)’,/,
           2:C’ -----’,17:C’ -----’,31:C’ -----’),
  OUTREC=(2:21,4,Y2V(-), * P’yymmdd’ -> C’yyyy-mm-dd’
          17:31,5, * Item code
          31:5,4,PD,M18,LENGTH=9, * Price
          120:X) * Pad with blanks
/*
```

Notice that **DATE=(4MD-)** tells **OUTFIL** to produce a **yyyy-mm-dd** format date in the title. In fact, **OUTFIL** lets you specify that the month (M), day (D), two-digit (Y) or four-digit year (4) and any separator character you select, be in any order you like in the report title.

Here’s what the report might look like.

Transaction Report for 1998-10-16

Date	Item Code	Price (\$)
-----	-----	-----
1999-11-21	ZZ723	1,203.25
1999-12-13	AX123	623.87
2000-01-08	BR321	318.05
2000-01-16	QZ520	1,621.36

## ICETOOL Reports

Transformed dates can easily be used in **ICETOOL reports** by using a **COPY or SORT operator** with **OUTFIL OUTREC** followed by a **DISPLAY operator**.

Here's an example of JCL and control statements to produce an ICETOOL report with a P'yymmdd' date field (starting in position 21) transformed into a C'yyyy-mm-dd' date field.

```
//REPORT2 EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//IN DD DSN=TRANS.INPUT,DISP=SHR
//TEMP DD DSN=&T1,DISP=(,PASS),SPACE=(TRK,(5,5)),UNIT=SYSDA
//RPT2 DD SYSOUT=*
//TOOLIN DD *
* Create temporary file with transformed date and
* other needed fields
  SORT FROM(IN) USING(TRAN)
* Create report
  DISPLAY FROM(TEMP) LIST(RPT2) BLANK -
    TITLE('Transaction Report for') DATE(4MD-) -
    HEADER('Date') ON(1,10,CH) -
    HEADER('Item Code') ON(15,5,CH) -
    HEADER('Price ($)') ON(25,4,PD,C1)
//TRANCNTL DD *
* Use century window of 1990-2089
  OPTION Y2PAST=1990
* Sort on P'yymmdd'
  SORT FIELDS=(21,4,Y2V,A)
* Transform P'yymmdd' (X'0yymmddC') to C'yyyy-mm-dd'
* and copy needed fields
  OUTFIL FNAMES=TEMP,
    OUTREC=(1:21,4,Y2V(-),          * P'yymmdd' -> C'yyyy-mm-dd'
           15:31,5,                 * Item code
           25:5,4)                  * Price
/*
```

Notice that **DATE(4MD-)** tells ICETOOL to produce a **yyyy-mm-dd** format date in the title. In fact, ICETOOL lets you specify that the month (M), day (D), two-digit (Y) or four-digit year (4) and any separator character you select, be in any order you like in the report title.

Here's what the report might look like.

```
Transaction Report for      1998-10-16

Date      Item Code      Price ($)
-----
1999-11-21  ZZ723      1,203.25
1999-12-13  AX123      623.87
2000-01-08  BR321      318.05
2000-01-16  QZ520      1,621.36
```

## Use COBOL MLE

With COBOL Millennium Language Extensions (MLE), it's easy to use DFSORT's Year 2000 features for sorting and merging. No need for JCL, SORT control statements or changes to your program logic. All you need to do is identify your date variables to the compiler with the new DATE FORMAT clause. If one of your SORT or MERGE verbs has a date field as an ASCENDING KEY or DESCENDING KEY, the COBOL MLE compiler will automatically build a SORT or MERGE control statement with the appropriate DFSORT Y2 format to order the date correctly using your century window.

With the latest MLE enhancements to the COBOL compilers, COBOL can automatically take advantage of DFSORT's new full date formats (Y2T, Y2U, Y2V, Y2W, Y2X and Y2Y) to sort and merge yyx...x and x...xyy dates and their special indicators (all zeros, all nines, SPACES, LOW-VALUES and HIGH-VALUES). These MLE enhancements are available:

- with APAR PQ22572 for the COBOL for MVS & VM compiler
- with APAR PQ22333 for the COBOL for OS/390 & VM compiler

COBOL MLE will also use DFSORT's year formats (Y2C, Y2Z, Y2S, Y2P, Y2D and Y2B) when appropriate. For example, the new enhancements now use DFSORT's Y2B format to support USAGE BINARY date fields.

## Use DFSORT's Year 2000 Features with COBOL

If you don't have COBOL MLE, but want to use DFSORT's Year 2000 with COBOL, you must **override** the DFSORT SORT statement generated by the compiler from the ASCENDING/DESCENDING keys specified in your COBOL program. You can find this generated SORT statement in DFSORT's message data set (normally SYSOUT). If the message data set isn't produced, add:

```
//SORTDIAG DD DUMMY
```

to your job. Override **all** of the fields in the generated SORT statement with your own SORT statement specified in the DFSPARM DD data set.

To set the fixed or sliding century window, supply a DFSORT OPTION control statement with the Y2PAST=x operand in the DFSPARM DD data set.

For example, if you had the following in your COBOL program:

```
SORT-ROUTINE SECTION.  
  SORT IN-FILE  
  ASCENDING DATEKEY, OTHERKEY  
  ...
```

where DATEKEY is of the form C'yymmdd' and OTHERKEY is a 10-byte character field, you might see the following **generated** SORT statement in DFSORT's message data set:

```
SORT FIELDS=(0011,0006,CH,A,0031,0010,CH,A)
```

The first field (11,6,CH,A) represents the DATEKEY field (11 is its starting position), and the second field (31,10,CH,A) represents the OTHERKEY field (31 is its starting position).

If you want to sort the DATEKEY field using a fixed century window of 1956-2055, just add the following to specify the century window and to override the DATEKEY field in the generated SORT statement with a Y2 field as explained in "Sort and Merge on Dates" on page 6.

```
//DFSPARM DD *
OPTION Y2PAST=1956      * set fixed CW of 1956-2055
SORT FIELDS=(11,6,Y2T,A, * sort C'yymmdd' as C'yyyymmdd' using CW
                 31,10,CH,A) * sort other key
```

## Multiple COBOL Sorts - DFSPARM

Using control statements in DFSPARM DD \* works fine when you only have **one** SORT verb in your COBOL program. But that won't work for **multiple** SORT verbs in your COBOL program, because the control statements in DFSPARM are normally used for every call to DFSORT (each SORT verb results in a call to DFSORT). In order to supply different SORT control statements for different COBOL SORT verbs, you must:

- place the control statements you want DFSORT to use for each SORT verb in a separate sequential data set or partitioned data set member
- specify these data sets (or members) in the same order as their SORT verbs
- specify FREE=CLOSE on each DFSPARM DD.

For example, if your COBOL program contained three SORT verbs, the following would cause the control statements in DP1 to be used for the first SORT verb, the control statements in DP2 to be used for the second SORT verb, and the control statements in DP3 to be used for the third SORT verb:

```
//DFSPARM DD DSN=DP1,DISP=SHR,FREE=CLOSE
//DFSPARM DD DSN=DP2,DISP=SHR,FREE=CLOSE
//DFSPARM DD DSN=DP3,DISP=SHR,FREE=CLOSE
```

Notes:

- JES3 does not allow the use of duplicate DFSPARM DD statements if the DD statements request JES3 or jointly-managed devices.
- DFSPARM is read by DFSORT, not COBOL, so this method can be used with any version of COBOL.
- DFSPARM may have been changed to another ddname at your installation via the ICEMAC PARMDDN=ddname operand. If so, use that ddname instead of DFSPARM.
- SYSOUT may have been changed to another ddname at your installation via the ICEMAC MSGDDN=ddname operand. If so, look for the generated SORT statement in the data set with that ddname.
- You can use ICETOOL's DEFAULTS operator to list all of your installation default values, including the ddnames for PARMDDN and MSGDDN.

## Multiple COBOL Sorts - SORT-CONTROL Special Register

You may not be able to use the DFSPARM technique described above in some situations such as with JES3 or when the COBOL sorts are not always executed in the same order. In these cases, you can use the following technique that takes advantage of COBOL's SORT-CONTROL special register if that feature is available.

Modify your COBOL source code to set the SORT-CONTROL special register to a different ddname before each SORT verb. For example, if you had two SORT verbs, you might specify:

```

MOVE 'SRT1DD' TO SORT-CONTROL.
SORT SORT-FILE1
...

MOVE 'SRT2DD' TO SORT-CONTROL.
SORT SORT-FILE2
...

```

Now you need some JCL for your control statements. Here's an example:

```

//SRT1DD DD *
  OPTION SORTDD=CTL1
/*
//SRT2DD DD *
  OPTION SORTDD=CTL2
/*
//CTL1CNTL DD *
  SORT FIELDS=(fields1)
  OPTION Y2PAST=x
/*
//CTL2CNTL DD *
  SORT FIELDS=(fields2)
  OPTION Y2PAST=y
/*

```

The first MOVE statement ties SRT1DD to the SORT verb for SORT-FILE1. The SRT1DD data set, in turn, has a SORTDD=CTL1 operand that ties the control statements in CTL1CNTL to the SORT verb for SORT-FILE1.

Likewise, the second MOVE statement ties SRT2DD to the SORT verb for SORT-FILE2. The SRT2DD data set, in turn, has a SORTDD=CTL2 operand that ties the control statements in CTL2CNTL to the SORT verb for SORT-FILE2.

With this method, you can specify the SORT verbs in any order. For example:

```

MOVE 'SRT2DD' TO SORT-CONTROL.
SORT SORT-FILE2
...

MOVE 'SRT1DD' TO SORT-CONTROL.
SORT SORT-FILE1
...

```

would still use the CTL1CNTL statements for the sort of SORT-FILE1 and the CTL2CNTL statements for the sort of SORT-FILE2. (Note that with the DFSPARM method, the wrong control statements would be used if you switched the SORT verbs like this without also switching the DFSPARM DD statements.)

Unfortunately, the SORT-CONTROL technique has the following drawbacks:

- You have to change your COBOL source to do the MOVES to SORT-CONTROL.
- If you are using SORTIN, SORTOUT and/or SORTWKxx, you will have to use ccccIN, ccccOUT and/or ccccWKxx instead. SORTDD=cccc has that effect.

Notes:

- The control statements in the data set pointed to by the SORT-CONTROL special register (for example, those in the SRT1DD data set) are added to the parameter list COBOL passes to DFSORT.

- The data set pointed to by the SORTDD=cccc operand (for example, the SRT1CNTL data set) is treated as a DFSORT SORTCNTL data set.

## SORT Examples (Order)

Listed here are examples of **SORT statements** for all kinds of dates. See “Sort and Merge on Dates” on page 6 for information on the date ordering techniques used in these examples.

To find an example for a particular date, look up the date (e.g. C’yymm’) in “Index into Examples” and use the index for the date (e.g. 05 for C’yymm’) to find the corresponding SORT statement in “Examples by Index” on page 26. See “Notes on Examples” on page 26 for important information about the examples.

### Index into Examples

(01) C’yymmdd’	(01) Z’yymmdd’	(02) P’yymmdd’	
(03) C’yyddd’	(03) Z’yyddd’	(04) P’yyddd’	
(05) C’yymm’	(05) Z’yymm’	(06) P’yymm’	
(07) C’yyq’	(07) Z’yyq’	(08) P’yyq’	
(09) C’mmddy’	(09) Z’mmddy’	(10) P’mmddy’	
(11) C’ddy’	(11) Z’ddy’	(12) P’ddy’	
(13) C’mmy’	(13) Z’mmy’	(14) P’mmy’	
(15) C’qyy’	(15) Z’qyy’	(16) P’qyy’	
(17) C’yy’	(17) Z’yy’	(18) P’yy’	
(19) X’yy’	(20) X’hh’ (binary)		
(21) C’yywwd’	(21) Z’yywwd’	(22) P’yywwd’	
(23) C’wwdy’	(23) Z’wwdy’	(24) P’wwdy’	
(25) C’yyddmm’	(25) Z’yyddmm’	(26) P’yyddmm’	
(27) C’ddmmy’	(27) Z’ddmmy’	(28) P’ddmmy’	
(29) C’yy/mm/dd’	(30) C’mm/dd/yy’	(31) C’yy/dd/mm’	(32) C’dd/mm/yy’
(33) C’yy/mm’	(34) C’mm/yy’		
(35) C’yyyymdd’	(35) Z’yyyymdd’	(36) P’yyyymdd’	
(37) C’mmddyyy’	(37) Z’mmddyyy’	(38) P’mmddyyy’	
(39) C’cyymmdd’	(39) Z’cyymmdd’	(40) P’cyymmdd’	
(41) C’mmddyyc’	(41) Z’mmddyyc’	(42) X’0mmddyyc’	
(43) X’yymmdd’	(44) X’mmddy’	(45) X’yyddmm’	(46) X’ddmmy’

### Notes on Examples

For the sake of simplicity, each example sorts on a single character, zoned decimal or packed decimal date field. In general, these fields can be represented as follows in hexadecimal:

CH/ZD: FdFd...sd  
PD: dd...ds

**d** is a decimal digit (0-9). **s** is a sign (0-F); it's ignored for sorting and merging.

Full date format and year format examples are shown as appropriate. **FDF** is used to denote full date format examples and **YRF** is used to denote year format examples. Note that the full date formats can handle all special indicators whereas the year formats are limited to the special indicators that Y2S can handle.

The elements of the various dates are as follows:

- **yy** - two-digit year
- **yyyy** - four-digit year
- **mm** - month
- **dd** - day of month
- **ddd** - day of year
- **q** - quarter
- **c** - century indicator - 0 for 19, 1 for 20, and so on
- **/** - separator (other characters can be used)
- **ww** - week
- **d** - day of week

The examples assume that the date field starts in position 1. For the year format examples, the relative position of each element is shown under each example using **p** and **p+n** notation, to make it easier to extrapolate the example to the general case.

## Examples by Index

### 1. C'yymmdd' or Z'yymmdd'

FDF: SORT FIELDS=(1,6,Y2T,A)

YRF: SORT FIELDS=(1,2,Y2C,A,3,4,BI,A)

yy at p, mm at p+2, dd at p+4

### 2. P'yymmdd' (X'0yymmdds')

FDF: SORT FIELDS=(1,4,Y2V,A)

YRF: SORT FIELDS=(1,2,Y2P,A,2,3,BI,A)

yy at p, mm at p+1, dd at p+2

### 3. C'yyddd' or Z'yyddd'

FDF: SORT FIELDS=(1,5,Y2T,A)

YRF: SORT FIELDS=(1,2,Y2C,A,3,3,BI,A)

yy at p, ddd at p+2

### 4. P'yyddd' (X'yyddd')

- FDF: SORT FIELDS=(1,3,Y2U,A)
- YRF: SORT FIELDS=(1,1,Y2D,A,2,2,BI,A)
- yy at p, ddd at p+1
5. C'yymm' or Z'yymm'
- FDF: SORT FIELDS=(1,4,Y2T,A)
- YRF: SORT FIELDS=(1,2,Y2C,A,3,2,BI,A)
- yy at p, mm at p+2
6. P'yymm' (X'0yymms')
- FDF: SORT FIELDS=(1,3,Y2V,A)
- YRF: SORT FIELDS=(1,2,Y2P,A,2,2,BI,A)
- yy at p, mm at p+1
7. C'yyq' or Z'yyq'
- FDF: SORT FIELDS=(1,3,Y2T,A)
- YRF: SORT FIELDS=(1,2,Y2C,A,3,1,BI,A)
- yy at p, q at p+2
8. P'yyq' (X'yyqs')
- FDF: SORT FIELDS=(1,2,Y2U,A)
- YRF: SORT FIELDS=(1,1,Y2D,A,2,1,BI,A)
- yy at p, q at p+1
9. C'mmddy' or Z'mmddy'
- FDF: SORT FIELDS=(1,6,Y2W,A)
- YRF: SORT FIELDS=(5,2,Y2C,A,1,4,BI,A)
- yy at p+4, mm at p, dd at p+2
10. P'mmddy' (X'0mmddyys')
- FDF: SORT FIELDS=(1,4,Y2Y,A)
- YRF: SORT FIELDS=(3,2,Y2P,A,1,3,BI,A)
- yy at p+2, mm at p, dd at p+1
11. C'dddy' or Z'dddy'
- FDF: SORT FIELDS=(1,5,Y2W,A)
- YRF: SORT FIELDS=(4,2,Y2C,A,1,3,BI,A)
- yy at p+3, ddd at p
12. P'dddy' (X'dddyys')

FDF: SORT FIELDS=(1,3,Y2X,A)

YRF: SORT FIELDS=(2,2,Y2P,A,1,2,BI,A)

yy at p+1, ddd at p

13. C'mmyy' or Z'mmyy'

FDF: SORT FIELDS=(1,4,Y2W,A)

YRF: SORT FIELDS=(3,2,Y2C,A,1,2,BI,A)

yy at p+2, mm at p

14. P'mmyy' (X'0mmyys')

FDF: SORT FIELDS=(1,3,Y2Y,A)

YRF: SORT FIELDS=(2,2,Y2P,A,1,2,BI,A)

yy at p+1, mm at p

15. C'qyy' or Z'qyy'

FDF: SORT FIELDS=(1,3,Y2W,A)

YRF: SORT FIELDS=(2,2,Y2C,A,1,1,BI,A)

yy at p+1, q at p

16. P'qyy' (X'qyys')

FDF: SORT FIELDS=(1,2,Y2X,A)

YRF: SORT FIELDS=(1,2,Y2P,A,1,1,BI,A)

yy at p, q at p

17. C'yy' or Z'yy'

YRF: SORT FIELDS=(1,2,Y2C,A)

18. P'yy' (X'0yys')

YRF: SORT FIELDS=(1,2,Y2P,A)

19. X'yy'

YRF: SORT FIELDS=(1,1,Y2D,A)

20. X'hh' (binary)

YRF: SORT FIELDS=(1,1,Y2B,A)

21. C'yywwd' or Z'yywwdd'

FDF: SORT FIELDS=(1,5,Y2T,A)

YRF: SORT FIELDS=(1,2,Y2C,A,3,3,BI,A)

yy at p, ww at p+2, d at p+4

22. P'yywwd' (X'yywwds')

- FDF: SORT FIELDS=(1,3,Y2U,A)
- YRF: SORT FIELDS=(1,1,Y2D,A,2,2,BI,A)  
yy at p, ww at p+1, d at p+2
23. C'wwdyy' or Z'wwdyy'  
FDF: SORT FIELDS=(1,5,Y2W,A)  
YRF: SORT FIELDS=(4,2,Y2C,A,1,3,BI,A)  
yy at p+3, ww at p, d at p+2
24. P'wwdyy' (X'wwdyys')  
FDF: SORT FIELDS=(1,3,Y2X,A)  
YRF: SORT FIELDS=(2,2,Y2P,A,1,2,BI,A)  
yy at p+1, ww at p, d at p+1
25. C'yyddmm' or Z'yyddmm'  
YRF: SORT FIELDS=(1,2,Y2C,A,5,2,BI,A,3,2,BI,A)  
yy at p, mm at p+4, dd at p+2
26. P'yyddmm' (X'0yyddmms')  
YRF: SORT FIELDS=(1,2,Y2P,A,3,2,PD0,A,2,2,BI,A)  
yy at p, mm at p+2, dd at p+1
27. C'ddmmyy' or Z'ddmmyy'  
YRF: SORT FIELDS=(5,2,Y2C,A,3,2,BI,A,1,2,BI,A)  
yy at p+4, mm at p+2, dd at p
28. P'ddmmyy' (X'0ddmmyys')  
YRF: SORT FIELDS=(3,2,Y2P,A,2,2,PD0,A,1,2,BI,A)  
yy at p+2, mm at p+1, dd at p
29. C'yy/mm/dd'  
YRF: SORT FIELDS=(1,2,Y2C,A,4,2,BI,A,7,2,BI,A)  
yy at p, mm at p+3, dd at p+6
30. C'mm/dd/yy'  
YRF: SORT FIELDS=(7,2,Y2C,A,1,2,BI,A,4,2,BI,A)  
yy at p+6, mm at p, dd at p+3
31. C'yy/dd/mm'  
YRF: SORT FIELDS=(1,2,Y2C,A,7,2,BI,A,4,2,BI,A)  
yy at p, mm at p+6, dd at p+3
32. C'dd/mm/yy'

YRF: SORT FIELDS=(7,2,Y2C,A,4,2,BI,A,1,2,BI,A)

yy at p+6, mm at p+3, dd at p

33. C'yy/mm'

YRF: SORT FIELDS=(1,2,Y2C,A,4,2,BI,A)

yy at p, mm at p+3

34. C'mm/yy'

YRF: SORT FIELDS=(4,2,Y2C,A,1,2,BI,A)

yy at p+2, mm at p

35. C'yyyymmdd' or Z'yyyymmdd'

SORT FIELDS=(1,8,BI,A)

yy at p, mm at p+4, dd at p+6

36. P'yyyymmdd' (X'0yyyymmdds')

SORT FIELDS=(1,5,BI,A)

yy at p, mm at p+2, dd at p+3

37. C'mmddyyyy' or Z'mmddyyyy'

SORT FIELDS=(5,4,BI,A,1,4,BI,A)

yy at p+4, mm at p, dd at p+2

38. P'mmddyyyy' (X'0mmddyyyys')

SORT FIELDS=(3,3,PD0,A,1,3,PD0,A)

yy at p+2, mm at p, dd at p+1

39. C'cyymmdd' or Z'cyymmdd'

SORT FIELDS=(1,7,BI,A)

c at p, yy at p+1, mm at p+3, dd at p+5

40. P'cyymmdd' (X'cyymmdds')

SORT FIELDS=(1,4,BI,A)

c at p, yy at p, mm at p+1, dd at p+2

41. C'mmddyyc' or Z'mmddyyc'

SORT FIELDS=(7,1,BI,A,5,2,BI,A,1,4,BI,A)

c at p+6, yy at p+4, mm at p, dd at p+2

42. X'0mmddyyc'

SORT FIELDS=(4.4,0.4,BI,A,3,2,PD0,A,1,3,PD0,A)

c at p+3.4, yy at p+2, mm at p, dd at p+1

43. X'yymmdd'

YRF: SORT FIELDS=(1,1,Y2D,A,2,2,BI,A)

yy at p, mm at p+1, dd at p+2

44. X'mmddy'

YRF: SORT FIELDS=(3,1,Y2D,A,1,2,BI,A)

yy at p+2, mm at p, dd at p+1

45. X'yyddmm'

YRF: SORT FIELDS=(1,1,Y2D,A,3,1,BI,A,2,1,BI,A)

yy at p, mm at p+2, dd at p+1

46. X'ddmmyy'

YRF: SORT FIELDS=(3,1,Y2D,A,2,1,BI,A,1,1,BI,A)

yy at p+2, mm at p+1, dd at p

## INCLUDE Examples (Compare)

Listed here are examples of **INCLUDE statements** for all kinds of dates. See “Compare Dates” on page 10 for information on the date comparison techniques used in these examples.

To find an example for a particular date, look up the date (e.g. C’yymm’) in “Index into Examples” and use the index for the date (e.g. 05 for C’yymm’) to find the corresponding **INCLUDE** statement in “Examples by Index” on page 33. See “Notes on Examples” for important information about the examples.

### Index into Examples

(01) C’yymmdd’	(01) Z’yymmdd’	(02) P’yymmdd’
(03) C’yyddd’	(03) Z’yyddd’	(04) P’yyddd’
(05) C’yymm’	(05) Z’yymm’	(06) P’yymm’
(07) C’yyq’	(07) Z’yyq’	(08) P’yyq’
(09) C’mmddy’	(09) Z’mmddy’	(10) P’mmddy’
(11) C’ddy’	(11) Z’ddy’	(12) P’ddy’
(13) C’mmy’	(13) Z’mmy’	(14) P’mmy’
(15) C’qyy’	(15) Z’qyy’	(16) P’qyy’
(17) C’yy’	(17) Z’yy’	(18) P’yy’
(19) X’yy’	(20) X’hh’ (binary)	
(21) C’yywwd’	(21) Z’yywwd’	(22) P’yywwd’
(23) C’wwdy’	(23) Z’wwdy’	(24) P’wwdy’
(25) C’yyyymmdd’	(26) Z’yyyymmdd’	(27) P’yyyymmdd’

### Notes on Examples

For the sake of simplicity, each example compares a single character, zoned decimal or packed decimal date field to a constant. In general, these fields can be represented as follows in hexadecimal:

```
CH/ZD: FdFd...sd
PD:    dd...dds
```

**d** is a decimal digit (0-9). **s** is a sign (0-F); it’s ignored for comparisons.

Full date format and year format examples are shown as appropriate. **FDF** is used to denote full date format examples and **YRF** is used to denote year format examples. Note that the full date formats can handle all special indicators whereas the year formats are limited to the special indicators that Y2S can handle.

**op** represents one of the following comparison operators: EQ, NE, GT, GE, LT, LE.

The elements of the various dates are as follows:

- **yy** - two-digit year

- **yyyy** - four-digit year
- **mm** - month
- **dd** - day of month
- **ddd** - day of year
- **q** - quarter
- **ww** - week
- **d** - day of week

The examples assume that the date field starts in position 1.

### Examples by Index

1. C'yymmdd' or Z'yymmdd'  
FDF: INCLUDE COND=(1,6,Y2T,op,Y' yymmdd')
2. P'yymmdd' (X'0yymmdds')  
FDF: INCLUDE COND=(1,4,Y2V,op,Y' yymmdd')
3. C'yyddd' or Z'yyddd'  
FDF: INCLUDE COND=(1,5,Y2T,op,Y' yyddd')
4. P'yyddd' (X'yyddd's')  
FDF: INCLUDE COND=(1,3,Y2U,op,Y' yyddd')
5. C'yymm' or Z'yymm'  
FDF: INCLUDE COND=(1,4,Y2T,op,Y' yymm')
6. P'yymm' (X'0yymms')  
FDF: INCLUDE COND=(1,3,Y2V,op,Y' yymm')
7. C'yyq' or Z'yyq'  
FDF: INCLUDE COND=(1,3,Y2T,op,Y' yyq')
8. P'yyq' (X'yyqs')  
FDF: INCLUDE COND=(1,2,Y2U,op,Y' yyq')
9. C'mmddy' or Z'mmddy'  
FDF: INCLUDE COND=(1,6,Y2W,op,Y' yymmdd')
10. P'mmddy' (X'0mmddy's')  
FDF: INCLUDE COND=(1,4,Y2Y,op,Y' yymmdd')
11. C'dddyy' or Z'dddyy'  
FDF: INCLUDE COND=(1,5,Y2W,op,Y' yyddd')
12. P'dddyy' (X'dddyy's')  
FDF: INCLUDE COND=(1,3,Y2X,op,Y' yyddd')
13. C'mmyy' or Z'mmyy'  
FDF: INCLUDE COND=(1,4,Y2W,op,Y' yymm')
14. P'mmyy' (X'0mmyy's')  
FDF: INCLUDE COND=(1,3,Y2Y,op,Y' yymm')

15. C'qyy' or Z'qyy'  
 FDF: INCLUDE COND=(1,3,Y2W,op,Y'yyq')
16. P'qyy' (X'qyys')  
 FDF: INCLUDE COND=(1,2,Y2X,op,Y'yyq')
17. C'yy' or Z'yy'  
 YRF: INCLUDE COND=(1,2,Y2C,op,Y'yy')
18. P'yy' (X'0yys')  
 YRF: INCLUDE COND=(1,2,Y2P,op,Y'yy')
19. X'yy'  
 YRF: INCLUDE COND=(1,1,Y2D,op,Y'yy')
20. X'hh' (binary)  
 YRF: INCLUDE COND=(1,1,Y2B,op,Y'yy')
21. C'yywwd' or Z'yywwdd'  
 FDF: INCLUDE COND=(1,5,Y2T,op,Y'yywwd')
22. P'yywwd' (X'yywwds')  
 FDF: INCLUDE COND=(1,3,Y2U,op,Y'yywwd')
23. C'wwdyy' or Z'wwdyy'  
 FDF: INCLUDE COND=(1,5,Y2W,op,Y'yywwd')
24. P'wwdyy' (X'wwdyys')  
 FDF: INCLUDE COND=(1,3,Y2X,op,Y'yywwd')
25. C'yyyymmdd'  
 INCLUDE COND=(1,8,BI,op,C'yyyymmdd')
26. Z'yyyymmdd'  
 INCLUDE COND=(1,8,ZD,op,yyyymmdd)
27. P'yyyymmdd' (X'0yyyymmdds')  
 INCLUDE COND=(1,5,PD,op,yyyymmdd)

## OUTFIL OUTREC Examples (Transform)

Listed here are examples of **OUTFIL OUTREC statements** for all kinds of dates. See “Transform Dates” on page 13 for information on the date transformation techniques used in these examples.

To find an example for a particular date, look up the date (e.g. C’yymm’) in “Index into Examples” and use the index for the date (e.g. 07 for C’yymm’) to find the corresponding OUTFIL statement in “Examples by Index” on page 36. See “Notes on Examples” on page 36 for important information about the examples.

### Index into Examples

(01) C’yymmdd’	(02) Z’yymmdd’	(03) P’yymmdd’	
(04) C’yyddd’	(05) Z’yyddd’	(06) P’yyddd’	
(07) C’yymm’	(08) Z’yymm’	(09) P’yymm’	
(10) C’yyq’	(11) Z’yyq’	(12) P’yyq’	
(13) C’mmddy’	(13) Z’mmddy’	(14) P’mmddy’	
(15) C’ddy’	(15) Z’ddy’	(16) P’ddy’	
(17) C’mmy’	(17) Z’mmy’	(18) P’mmy’	
(19) C’qyy’	(19) Z’qyy’	(20) P’qyy’	
(21) C’yy’	(21) Z’yy’	(22) P’yy’	
(23) X’yy’	(24) X’hh’ (binary)		
(25) C’yywwd’	(26) Z’yywwd’	(27) P’yywwd’	
(28) C’wwdy’	(28) Z’wwdy’	(29) P’wwdy’	
(30) C’yyddmm’	(31) Z’yyddmm’	(32) P’yyddmm’	
(33) C’ddmmy’	(33) Z’ddmmy’	(34) P’ddmmy’	
(35) C’yy/mm/dd’	(36) C’mm/dd/yy’	(37) C’yy/dd/mm’	(38) C’dd/mm/yy’
(39) C’yy/mm’	(40) C’mm/yy’		
(41) C’yyyymdd’	(42) Z’yyyymdd’	(43) P’yyyymdd’	
(44) C’mmddyyy’	(45) Z’mmddyyy’	(46) P’mmddyyy’	
(47) C’cyymmdd’	(48) Z’cyymmdd’	(49) P’cyymmdd’	
(50) C’mmddyyc’	(51) X’0mmddyyc’		
(52) X’yymmdd’	(53) X’mmddy’	(54) X’yyddmm’	(55) X’ddmmy’

### Notes on Examples

For the sake of simplicity, each example transforms a single character, zoned decimal or packed decimal date field to a four-digit year date field. In general, these fields can be represented as follows in hexadecimal:

CH: FdFd...Fd  
ZD: FdFd...sd  
PD: dd...ds

**d** is a decimal digit (0-9). **s** is the sign (0-F) for ZD and PD values. When you use year formats to transform dates, the sign and the individual non-year parts (e.g. month and day) can require special handling, such as the use of ZD, PD or PD0 formats and the M11 edit mask, to make the output printable. When you use full date formats to transform dates, DFSORT automatically takes care of these complexities for you.

Examples of transformation of dates to character with and without separators and packed decimal forms are shown, using full date formats and year formats, as appropriate. **FDF** is used to denote full date format examples and **YRF** is used to denote year format examples. Note that the full date formats can handle all special indicators whereas the year formats are limited to the special indicators that Y2S can handle.

The elements of the various dates are as follows:

- **yy** - two-digit year
- **yyyy** - four-digit year
- **mm** - month
- **dd** - day of month
- **ddd** - day of year
- **q** - quarter
- **c** - century indicator - 0 for 19, 1 for 20, and so on
- **/** - separator (other characters can be used)
- **ww** - week
- **d** - day of week

The examples assume the following:

- The input date field starts in position 1. For the year format examples, the relative position of each element is shown under each example using **p** and **p+n** notation, to make it easier to extrapolate the example to the general case.
- The input date field is to be transformed to the **output date** field shown before each example. You can use the examples as a guide to producing any output date form you like.
- The output date field is to start in position 1. You can use the column operand (c:) to place the output date field in any position you like.

## Examples by Index

1. C'yymmdd' to C'yyyymmdd'

FDF: OUTFIL OUTREC=(1,6,Y2T)

YRF: OUTFIL OUTREC=(1,2,Y2C,3,4)

yy at p, mm at p+2, dd at p+4

C'yymmdd' to C'yyyy/mm/dd'

FDF: OUTFIL OUTREC=(1,6,Y2T(/))

YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,2,C'/',5,2)  
 yy at p, mm at p+2, dd at p+4

C'yymmdd' to P'yyyyymmdd' (X'0yyyyymmddC')

FDF: OUTFIL OUTREC=(1,6,Y2TP)

2. Z'yymmdd' to C'yyyyymmdd'

FDF: OUTFIL OUTREC=(1,6,Y2T)

YRF: OUTFIL OUTREC=(1,2,Y2C,3,2,5,2,ZD,M11)  
 yy at p, mm at p+2, dd at p+4

Z'yymmdd' to C'yyyy/mm/dd'

FDF: OUTFIL OUTREC=(1,6,Y2T(/))

YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,2,C'/',5,2,ZD,M11)  
 yy at p, mm at p+2, dd at p+4

Z'yymmdd' to P'yyyyymmdd' (X'0yyyyymmddC')

FDF: OUTFIL OUTREC=(1,6,Y2TP)

3. P'yymmdd' (X'0yymmdds') to C'yyyyymmdd'

FDF: OUTFIL OUTREC=(1,4,Y2V)

YRF: OUTFIL OUTREC=(1,2,Y2P,2,3,P0,M11)  
 yy at p, mm at p+1, dd at p+2

P'yymmdd' (X'0yymmdds') to C'yyyy/mm/dd'

FDF: OUTFIL OUTREC=(1,4,Y2V(/))

YRF: OUTFIL OUTREC=(1,2,Y2P,C'/',2,2,P0,M11,C'/',3,2,P0,M11)  
 yy at p, mm at p+1, dd at p+2

P'yymmdd' (X'0yymmdds') to P'yyyyymmdd' (X'0yyyyymmddC')

FDF: OUTFIL OUTREC=(1,4,Y2VP)

4. C'yyddd' to C'yyyyddd'

FDF: OUTFIL OUTREC=(1,5,Y2T)

YRF: OUTFIL OUTREC=(1,2,Y2C,3,3)  
 yy at p, ddd at p+2

C'yyddd' to C'yyyy/ddd'

FDF: OUTFIL OUTREC=(1,5,Y2T(/))

YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,3)  
 yy at p, ddd at p+2

C'yyddd' to P'yyyyddd' (X'yyyydddC')

FDF:  OUTFIL  OUTREC=(1,5,Y2TP)

5. Z'yyddd' to C'yyyyddd'

FDF:  OUTFIL  OUTREC=(1,5,Y2T)

YRF:  OUTFIL  OUTREC=(1,2,Y2C,3,3,ZD,M11)

yy at p, ddd at p+2

Z'yyddd' to C'yyyy/ddd'

FDF:  OUTFIL  OUTREC=(1,5,Y2T(/))

YRF:  OUTFIL  OUTREC=(1,2,Y2C,C'/',3,3,ZD,M11)

yy at p, ddd at p+2

Z'yyddd' to P'yyyyddd' (X'yyyydddC')

FDF:  OUTFIL  OUTREC=(1,5,Y2TP)

6. P'yyddd' (X'yyddd') to C'yyyyddd'

FDF:  OUTFIL  OUTREC=(1,3,Y2U)

YRF:  OUTFIL  OUTREC=(1,1,Y2D,2,2,PD,M11)

yy at p, ddd at p+1

P'yyddd' (X'yyddd') to C'yyyy/ddd'

FDF:  OUTFIL  OUTREC=(1,3,Y2U(/))

YRF:  OUTFIL  OUTREC=(1,1,Y2D,C'/',2,2,PD,M11)

yy at p, ddd at p+1

P'yyddd' (X'yyddd') to P'yyyyddd' (X'yyyydddC')

FDF:  OUTFIL  OUTREC=(1,3,Y2UP)

7. C'yymm' to C'yyyyymm'

FDF:  OUTFIL  OUTREC=(1,4,Y2T)

YRF:  OUTFIL  OUTREC=(1,2,Y2C,3,2)

yy at p, mm at p+2

C'yymm' to C'yyyy/mm'

FDF:  OUTFIL  OUTREC=(1,4,Y2T(/))

YRF:  OUTFIL  OUTREC=(1,2,Y2C,C'/',3,2)

yy at p, mm at p+2

C'yymm' to P'yyyyymm' (X'0yyyyymmC')

FDF:  OUTFIL  OUTREC=(1,4,Y2TP)

8. Z'yymm' to C'yyyyymm'

FDF: OUTFIL OUTREC=(1,4,Y2T)

YRF: OUTFIL OUTREC=(1,2,Y2C,3,2,ZD,M11)

yy at p, mm at p+2

Z'yymm' to C'yyyy/mm'

FDF: OUTFIL OUTREC=(1,4,Y2T(/))

YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,2,ZD,M11)

yy at p, mm at p+2

Z'yymm' to P'yyyyymm' (X'0yyyyymmC')

FDF: OUTFIL OUTREC=(1,4,Y2TP)

9. P'yymm' (X'0yymms') to C'yyyyymm'

FDF: OUTFIL OUTREC=(1,3,Y2V)

YRF: OUTFIL OUTREC=(1,2,Y2P,2,2,PDO,M11)

yy at p, mm at p+1

P'yymm' (X'0yymms') to C'yyyy/mm'

FDF: OUTFIL OUTREC=(1,3,Y2V(/))

YRF: OUTFIL OUTREC=(1,2,Y2P,C'/',2,2,PDO,M11)

yy at p, mm at p+1

P'yymm' (X'0yymms') to P'yyyyymm' (X'0yyyyymmC')

FDF: OUTFIL OUTREC=(1,3,Y2VP)

10. C'yyq' to C'yyyyq'

FDF: OUTFIL OUTREC=(1,3,Y2T)

YRF: OUTFIL OUTREC=(1,2,Y2C,3,1)

yy at p, q at p+2

C'yyq' to C'yyyy/q'

FDF: OUTFIL OUTREC=(1,3,Y2T(/))

YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,1)

yy at p, q at p+2

C'yyq' to P'yyyyq' (X'yyyyqC')

FDF: OUTFIL OUTREC=(1,3,Y2TP)

11. Z'yyq' to C'yyyyq'

FDF: OUTFIL OUTREC=(1,3,Y2T)

YRF: OUTFIL OUTREC=(1,2,Y2C,3,1,ZD,M11)

yy at p, q at p+2

Z'yyq' to C'yyyy/q'

FDF: OUTFIL OUTREC=(1,3,Y2T(/))

YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,1,ZD,M11)

yy at p, q at p+2

Z'yyq' to P'yyyyq' (X'yyyyqC')

FDF: OUTFIL OUTREC=(1,3,Y2TP)

12. P'yyq' (X'yyqs') to C'yyyyq'

FDF: OUTFIL OUTREC=(1,2,Y2U)

YRF: OUTFIL OUTREC=(1,1,Y2D,2,1,PD,M11)

yy at p, q at p+1

P'yyq' (X'yyqs') to C'yyyy/q'

FDF: OUTFIL OUTREC=(1,2,Y2U(/))

YRF: OUTFIL OUTREC=(1,1,Y2D,C'/',2,1,PD,M11)

yy at p, q at p+1

P'yyq' (X'yyqs') to P'yyyyq' (X'yyyyqC')

FDF: OUTFIL OUTREC=(1,2,Y2UP)

13. C'mmddy' or Z'mmddy' to C'mmddyyy'

FDF: OUTFIL OUTREC=(1,6,Y2W)

YRF: OUTFIL OUTREC=(1,4,5,2,Y2C)

mm at p, dd at p+2, yy at p+4

C'mmddy' or Z'mmddy' to C'mm/dd/yyyy'

FDF: OUTFIL OUTREC=(1,6,Y2W(/))

YRF: OUTFIL OUTREC=(1,2,C'/',3,2,C'/',5,2,Y2C)

mm at p, dd at p+2, yy at p+4

C'mmddy' or Z'mmddy' to P'mmddyyy' (X'0mmddyyyC')

FDF: OUTFIL OUTREC=(1,6,Y2WP)

14. P'mmddy' (X'0mmddyys') to C'mmddyyy'

FDF: OUTFIL OUTREC=(1,4,Y2Y)

YRF: OUTFIL OUTREC=(1,3,PD0,M11,3,2,Y2P)

mm at p, dd at p+1, yy at p+2

P'mmddy' (X'0mmddyys') to C'mm/dd/yyyy'

FDF: OUTFIL OUTREC=(1,4,Y2Y(/))

YRF: OUTFIL OUTREC=(1,2,PD0,M11,C'/',2,2,PD0,M11,C'/',3,2,Y2P)

mm at p, dd at p+1, yy at p+2

P'mmddy' (X'0mmddyys') to P'mmddyyy' (X'0mmddyyyC')

- FDF: OUTFIL OUTREC=(1,4,Y2YP)
15. C'dddy' or Z'dddy' to C'ddyyy'
- FDF: OUTFIL OUTREC=(1,5,Y2W)
- YRF: OUTFIL OUTREC=(1,3,4,2,Y2C)
- ddd at p, yy at p+3
- C'dddy' or Z'dddy' to C'dd/yyyy'
- FDF: OUTFIL OUTREC=(1,5,Y2W(/))
- YRF: OUTFIL OUTREC=(1,3,C'/',4,2,Y2C)
- ddd at p, yy at p+3
- C'dddy' or Z'dddy' to P'ddyyy' (X'ddyyyC')
- FDF: OUTFIL OUTREC=(1,5,Y2WP)
16. P'dddy' (X'ddyy's) to C'ddyyy'
- FDF: OUTFIL OUTREC=(1,3,Y2X)
- YRF: OUTFIL OUTREC=(1,2,PD,M11,2,2,Y2P)
- ddd at p, yy at p+1
- P'dddy' (X'ddyy's) to C'dd/yyyy'
- FDF: OUTFIL OUTREC=(1,3,Y2X(/))
- YRF: OUTFIL OUTREC=(1,2,PD,M11,C'/',2,2,Y2P)
- ddd at p, yy at p+1
- P'dddy' (X'ddyy's) to P'ddyyy' (X'ddyyyC')
- FDF: OUTFIL OUTREC=(1,3,Y2XP)
17. C'mmy' or Z'mmy' to C'mmyy'
- FDF: OUTFIL OUTREC=(1,4,Y2W)
- YRF: OUTFIL OUTREC=(1,2,3,2,Y2C)
- mm at p, yy at p+2
- C'mmy' or Z'mmy' to C'mm/yyyy'
- FDF: OUTFIL OUTREC=(1,4,Y2W(/))
- YRF: OUTFIL OUTREC=(1,2,C'/',3,2,Y2C)
- mm at p, yy at p+2
- C'mmy' or Z'mmy' to P'mmyy' (X'0mmyyC')
- FDF: OUTFIL OUTREC=(1,4,Y2WP)
18. P'mmy' (X'0mmy's) to C'mmyy'

FDF: OUTFIL OUTREC=(1,3,Y2Y)

YRF: OUTFIL OUTREC=(1,2,PD0,M11,2,2,Y2P)

mm at p, yy at p+1

P'mmyy' (X'0mmyys') to C'mm/yyyy'

FDF: OUTFIL OUTREC=(1,3,Y2Y(/))

YRF: OUTFIL OUTREC=(1,2,PD0,M11,C'/',2,2,Y2P)

mm at p, yy at p+1

P'mmyy' (X'0mmyys') to P'mmyyyy' (X'0mmyyyyC')

FDF: OUTFIL OUTREC=(1,3,Y2YP)

19. C'qyy' or Z'qyy' to C'qyyy'

FDF: OUTFIL OUTREC=(1,3,Y2W)

YRF: OUTFIL OUTREC=(1,1,2,2,Y2C)

q at p, yy at p+1

C'qyy' or Z'qyy' to C'q/yyyy'

FDF: OUTFIL OUTREC=(1,3,Y2W(/))

YRF: OUTFIL OUTREC=(1,1,C'/',2,2,Y2C)

q at p, yy at p+1

C'qyy' or Z'qyy' to P'qyyyy' (X'qyyyyC')

FDF: OUTFIL OUTREC=(1,3,Y2WP)

20. P'qyy' (X'qyys') to C'qyyy'

FDF: OUTFIL OUTREC=(1,2,Y2X)

YRF: OUTFIL OUTREC=(1,1,PD,M11,1,2,Y2P)

q at p, yy at p+1

P'qyy' (X'qyys') to C'q/yyyy'

FDF: OUTFIL OUTREC=(1,2,Y2X(/))

YRF: OUTFIL OUTREC=(1,1,PD,M11,C'/',1,2,Y2P)

q at p, yy at p+1

P'qyy' (X'qyys') to P'qyyyy' (X'qyyyyC')

FDF: OUTFIL OUTREC=(1,2,Y2XP)

21. C'yy' or Z'yy' to C'yyyy'

YRF: OUTFIL OUTREC=(1,2,Y2C)

22. P'yy' (X'0yys') to C'yyyy'

YRF: OUTFIL OUTREC=(1,2,Y2P)

P'yy' (X'0yys') to P'yyyy' (X'0yyyyC')

YRF: OUTFIL OUTREC=(1,2,Y2PP)

23. X'yy' to C'yyyy'  
 YRF: OUTFIL OUTREC=(1,1,Y2D)  
 X'yy' to X'yyyy'  
 YRF: OUTFIL OUTREC=(1,1,Y2DP)
24. X'hh' (binary) to C'yyyy'  
 YRF: OUTFIL OUTREC=(1,1,Y2B)
25. C'yywwd' to C'yyyywwd'  
 FDF: OUTFIL OUTREC=(1,5,Y2T)  
 YRF: OUTFIL OUTREC=(1,2,Y2C,3,3)  
 yy at p, ww at p+2, d at p+4  
 C'yywwd' to C'yyyy/ww/d'  
 YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,2,C'/',5,1)  
 yy at p, ww at p+2, d at p+4  
 C'yywwd' to P'yyyywwd' (X'yyyywwdC')  
 FDF: OUTFIL OUTREC=(1,5,Y2TP)
26. Z'yywwd' to C'yyyywwd'  
 FDF: OUTFIL OUTREC=(1,5,Y2T)  
 YRF: OUTFIL OUTREC=(1,2,Y2C,3,2,5,1,ZD,M11)  
 yy at p, ww at p+2, d at p+4  
 Z'yywwd' to C'yyyy/ww/d'  
 YRF: OUTFIL OUTREC=(1,2,Y2C,C'/',3,2,C'/',5,1,ZD,M11)  
 yy at p, ww at p+2, d at p+4  
 Z'yywwd' to P'yyyywwd' (X'yyyywwdC')  
 FDF: OUTFIL OUTREC=(1,5,Y2TP)
27. P'yywwd' (X'yywwds') to C'yyyywwd'  
 FDF: OUTFIL OUTREC=(1,3,Y2U)  
 YRF: OUTFIL OUTREC=(1,1,Y2D,2,2,PD,M11)  
 yy at p, ww at p+1, d at p+2  
 P'yywwd' (X'yywwds') to C'yyyy/ww/d'  
 YRF: OUTFIL OUTREC=(1,1,Y2D,C'/',2,1,HEX,C'/',3,1,PD,M11)  
 yy at p, ww at p+1, d at p+2  
 P'yywwd' (X'yywwds') to P'yyyywwd' (X'yyyywwdC')  
 FDF: OUTFIL OUTREC=(1,3,Y2UP)
28. C'wwdyy' or Z'wwdyy' to C'wwdyyyy'

FDF: UTFIL OUTREC=(1,5,Y2W)

YRF: UTFIL OUTREC=(1,3,4,2,Y2C)

ww at p, d at p+2, yy at p+3

C'wwdy' or Z'wwdy' to C'ww/d/yyyy'

YRF: UTFIL OUTREC=(1,2,C'/',3,1,C'/',4,2,Y2C)

ww at p, d at p+2, yy at p+3

C'wwdy' or Z'wwdy' to P'wwdy' (X'wwdyC')

FDF: UTFIL OUTREC=(1,5,Y2WP)

29. P'wwdy' (X'wwdy') to C'wwdy'

FDF: UTFIL OUTREC=(1,3,Y2X)

YRF: UTFIL OUTREC=(1,2,PD,M11,2,2,Y2P)

ww at p, d at p+1, yy at p+1

P'wwdy' (X'wwdy') to C'ww/d/yyyy'

YRF: UTFIL OUTREC=(1,1,HEX,C'/',2,1,PD,M11,C'/',2,2,Y2P)

ww at p, d at p+1, yy at p+1

P'wwdy' (X'wwdy') to P'wwdy' (X'wwdyC')

FDF: UTFIL OUTREC=(1,3,Y2XP)

30. C'yyddmm' to C'yyyddmm'

YRF: UTFIL OUTREC=(1,2,Y2C,3,4)

yy at p, dd at p+2, mm at p+4

C'yyddmm' to C'yyy/dd/mm'

YRF: UTFIL OUTREC=(1,2,Y2C,C'/',3,2,C'/',5,2)

yy at p, dd at p+2, mm at p+4

31. Z'yyddmm' to C'yyyddmm'

YRF: UTFIL OUTREC=(1,2,Y2C,3,2,5,2,ZD,M11)

yy at p, dd at p+2, mm at p+4

Z'yyddmm' to C'yyy/dd/mm'

YRF: UTFIL OUTREC=(1,2,Y2C,C'/',3,2,C'/',5,2,ZD,M11)

yy at p, dd at p+2, mm at p+4

32. P'yyddmm' (X'0yyddmms') to C'yyyddmm'

YRF: UTFIL OUTREC=(1,2,Y2P,2,3,PD0,M11)

yy at p, dd at p+1, mm at p+2

P'yyddmm' (X'0yyddmms') to C'yyy/dd/mm'

YRF: UTFIL OUTREC=(1,2,Y2P,C'/',2,2,PD0,M11,C'/',3,2,PD0,M11)

yy at p, dd at p+1, mm at p+2

33. C'ddmmyy' or Z'ddmmyy' to C'ddmmyyyy'  
 YRF: OUTFIL OUTREC=(1,4,5,2,Y2C)  
 dd at p, mm at p+2, yy at p+4  
 C'ddmmyy' or Z'ddmmyy' to C'dd/mm/yyyy'  
 YRF: OUTFIL OUTREC=(1,2,C'/',3,2,C'/',5,2,Y2C)  
 dd at p, mm at p+2, yy at p+4
34. P'ddmmyy' (X'0ddmmys') to C'ddmmyyyy'  
 YRF: OUTFIL OUTREC=(1,3,P0,M11,3,2,Y2P)  
 dd at p, mm at p+1, yy at p+2  
 P'ddmmyy' (X'0ddmmys') to C'dd/mm/yyyy'  
 YRF: OUTFIL OUTREC=(1,2,P0,M11,C'/',2,2,P0,M11,C'/',3,2,Y2P)  
 dd at p, mm at p+1, yy at p+2
35. C'yy/mm/dd' to C'yyyymmdd'  
 YRF: OUTFIL OUTREC=(1,2,Y2C,4,2,7,2)  
 yy at p, mm at p+3, dd at p+6  
 C'yy/mm/dd' to C'yyyy/mm/dd'  
 YRF: OUTFIL OUTREC=(1,2,Y2C,3,6)  
 yy at p, mm at p+3, dd at p+6
36. C'mm/dd/yy' to C'mmddyyyy'  
 YRF: OUTFIL OUTREC=(1,2,4,2,7,2,Y2C)  
 mm at p, dd at p+3, yy at p+6  
 C'mm/dd/yy' to C'mm/dd/yyyy'  
 YRF: OUTFIL OUTREC=(1,6,7,2,Y2C)  
 mm at p, dd at p+3, yy at p+6
37. C'yy/dd/mm' to C'yyyddmm'  
 YRF: OUTFIL OUTREC=(1,2,Y2C,4,2,7,2)  
 yy at p, mm at p+3, mm at p+6  
 C'yy/dd/mm' to C'yyyy/dd/mm'  
 YRF: OUTFIL OUTREC=(1,2,Y2C,3,6)  
 yy at p, mm at p+3, mm at p+6
38. C'dd/mm/yy' to C'ddmmyyyy'  
 YRF: OUTFIL OUTREC=(1,2,4,2,7,2,Y2C)  
 dd at p, mm at p+3, yy at p+6  
 C'dd/mm/yy' to C'dd/mm/yyyy'

YRF: UTFIL OUTREC=(1,6,7,2,Y2C)

dd at p, mm at p+3, yy at p+6

39. C'yy/mm' to C'yyyymm'

YRF: UTFIL OUTREC=(1,2,Y2C,4,2)

yy at p, mm at p+3

C'yy/mm' to C'yyyy/mm'

YRF: UTFIL OUTREC=(1,2,Y2C,3,3)

yy at p, mm at p+3

40. C'mm/yy' to C'mmyyyy'

YRF: UTFIL OUTREC=(1,2,4,2,Y2C)

mm at p, yy at p+3

C'mm/yy' to C'mm/yyyy'

YRF: UTFIL OUTREC=(1,3,4,2,Y2C)

mm at p, yy at p+3

41. C'yyyymmdd' to C'yyyymmdd'

UTFIL OUTREC=(1,8)

yy at p, mm at p+4, dd at p+6

C'yyyymmdd' to C'yyyy/mm/dd'

UTFIL OUTREC=(1,4,C'/',5,2,C'/',7,2)

yy at p, mm at p+4, dd at p+6

42. Z'yyyymmdd' to C'yyyymmdd'

UTFIL OUTREC=(1,8,ZD,M11)

yy at p, mm at p+4, dd at p+6

Z'yyyymmdd' to C'yyyy/mm/dd'

UTFIL OUTREC=(1,4,C'/',5,2,C'/',7,2,ZD,M11)

yy at p, mm at p+4, dd at p+6

43. P'yyyymmdd' (X'0yyyymmdds') to C'yyyymmdd'

UTFIL OUTREC=(1,5,PDO,M11)

yy at p, mm at p+2, dd at p+3

P'yyyymmdd' (X'0yyyymmdds') to C'yyyy/mm/dd'

UTFIL OUTREC=(1,3,PDO,M11,C'/',3,2,PDO,M11,C'/',4,2,PDO,M11)

yy at p, mm at p+2, dd at p+3

44. C'mmddyyyy' to C'mmddyyyy'

UTFIL OUTREC=(1,8)

mm at p, dd at p+2, yy at p+4

C'mmddyyyy' to C'mm/dd/yyyy'

OUTFIL OUTREC=(1,2,C'/',3,2,C'/',5,4)

mm at p, dd at p+2, yy at p+4

45. Z'mmddyyyy' to C'mmddyyyy'

OUTFIL OUTREC=(1,8,ZD,M11)

mm at p, dd at p+2, yy at p+4

Z'mmddyyyy' to C'mm/dd/yyyy'

OUTFIL OUTREC=(1,2,C'/',3,2,C'/',5,4,ZD,M11)

mm at p, dd at p+2, yy at p+4

46. P'mmddyyyy' (X'0mmddyyyys') to C'mmddyyyy'

OUTFIL OUTREC=(1,5,PDO,M11)

mm at p, dd at p+1, yy at p+2

P'mmddyyyy' (X'0mmddyyyys') to C'mm/dd/yyyy'

OUTFIL OUTREC=(1,2,PDO,M11,C'/',2,2,PDO,M11,C'/',3,3,PDO,M11)

mm at p, dd at p+1, yy at p+2

47. C'cyymmdd' to C'yyyymmdd'

OUTFIL OUTREC=(1,1,CHANGE=(2,  
C'0',C'19',  
C'1',C'20',  
C'2',C'21'),  
2,6)

c at p, yy at p+1, mm at p+3, dd at p+5

C'cyymmdd' to C'yyyy/mm/dd'

OUTFIL OUTREC=(1,1,CHANGE=(2,  
C'0',C'19',  
C'1',C'20',  
C'2',C'21'),  
2,2,C'/',4,2,C'/',6,2)

c at p, yy at p+1, mm at p+3, dd at p+5

48. Z'cyymmdd' to C'yyyymmdd'

OUTFIL OUTREC=(1,1,CHANGE=(2,  
C'0',C'19',  
C'1',C'20',  
C'2',C'21'),  
2,4,6,2,ZD,M11)

c at p, yy at p+1, mm at p+3, dd at p+5

Z'cyymmdd' to C'yyyy/mm/dd'

```

OUTFIL OUTREC=(1,1,CHANGE=(2,
                  C'0',C'19',
                  C'1',C'20',
                  C'2',C'21'),
                2,2,C'/',4,2,C'/',6,2,ZD,M11)

```

c at p, yy at p+1, mm at p+3, dd at p+5

49. P'cyymmdd' (X'cyymmdds') to C'yyyymmdd'

```

OUTFIL OUTREC=(1,1,CHANGE=(2,
                  B'0000....',C'19',
                  B'0001....',C'20',
                  B'0010....',C'21'),
                1,4,PDO,M11)

```

c at p, yy at p, mm at p+1, dd at p+2

P'cyymmdd' (X'cyymmdds') to C'yyyy/mm/dd'

```

OUTFIL OUTREC=(1,1,CHANGE=(2,
                  B'0000....',C'19',
                  B'0001....',C'20',
                  B'0010....',C'21'),
                1,2,PDO,M11,C'/',2,2,PDO,M11,C'/',3,2,PDO,M11)

```

c at p, yy at p, mm at p+1, dd at p+2

50. C'mmddyyc' to C'mmddyyyy'

```

OUTFIL OUTREC=(1,4,
                7,1,CHANGE=(2,
                  C'0',C'19',
                  C'1',C'20',
                  C'2',C'21'),
                5,2)

```

mm at p, dd at p+2, yy at p+4, c at p+6

C'mmddyyc' to C'mm/dd/yyyy'

```

OUTFIL OUTREC=(1,2,C'/',3,2,C'/',
                7,1,CHANGE=(2,
                  C'0',C'19',
                  C'1',C'20',
                  C'2',C'21'),
                5,2)

```

mm at p, dd at p+2, yy at p+4, c at p+6

51. X'0mmddyyc' to C'mmddyyyy'

```

OUTFIL OUTREC=(1,3,PDO,M11,
                4,1,CHANGE=(2,
                  B'....0000',C'19',
                  B'....0001',C'20',
                  B'....0010',C'21'),
                3,2,PDO,M11)

```

mm at p, dd at p+1, yy at p+2, c at p+3

X'0mmddyyc' to C'mm/dd/yyyy'

```
OUTFIL OUTREC=(1,2,PDO,M11,C'/',2,2,PDO,M11,C'/',
4,1,CHANGE=(2,
B'...0000',C'19',
B'...0001',C'20',
B'...0010',C'21'),
3,2,PDO,M11)
```

mm at p, dd at p+1, yy at p+2, c at p+3

52. X'yymmdd' to C'yyyymmdd'

YRF: OUTFIL OUTREC=(1,1,Y2D,2,2,HEX)

yy at p, mm at p+1, dd at p+2

X'yymmdd' to C'yyyy/mm/dd'

YRF: OUTFIL OUTREC=(1,1,Y2D,C'/',2,1,HEX,C'/',3,1,HEX)

yy at p, mm at p+1, dd at p+2

53. X'mmddy' to C'mmddyyy'

YRF: OUTFIL OUTREC=(1,2,HEX,3,1,Y2D)

mm at p, dd at p+1, yy at p+2

X'mmddy' to C'mm/dd/yyyy'

YRF: OUTFIL OUTREC=(1,1,HEX,C'/',2,1,HEX,C'/',3,1,Y2D)

mm at p, dd at p+1, yy at p+2

54. X'yyddmm' to C'yyyddmm'

YRF: OUTFIL OUTREC=(1,1,Y2D,2,2,HEX)

yy at p, dd at p+1, mm at p+2

X'yyddmm' to C'yyyy/dd/mm'

YRF: OUTFIL OUTREC=(1,1,Y2D,C'/',2,1,HEX,C'/',3,1,HEX)

yy at p, dd at p+1, mm at p+2

55. X'ddmmyy' to C'ddmmyyyy'

YRF: OUTFIL OUTREC=(1,2,HEX,3,1,Y2D)

dd at p, mm at p+1, yy at p+2

X'ddmmyy' to C'dd/mm/yyyy'

YRF: OUTFIL OUTREC=(1,1,HEX,C'/',2,1,HEX,C'/',3,1,Y2D)

dd at p, mm at p+1, yy at p+2