

Date Manipulation Routines:

Central to any leasing, mortgage, investment or general financial application is the manipulation of dates. This includes date-dependent calculations, "future" date computations, and special formatting for reports.

The routines included here represent those that are (a) self-contained, and (b) fully Y2K compliant.

They are listed in alphabetical order by routine **name**.

DTDADD — Subroutine {COBOL}

Purpose: Adds an unsigned number of **days** to an input date in 8-digit format with 3 components, i.e. year-month-day as "CCYYMMDD", then returns resultant date (separate variable) in same format. Takes actual calendar into account.

Usage: Computing "future" dates (commencement date, due date, return date, etc.) based upon a known "current" date and a "number of calendar days" from that point.

Applications include leasing, mortgages, interest, investments, billings/collections, and any other financial applications; also estimated mail/shipping delivery dates.

Calling Sequence:

```
CALL 'DTDADD' USING  INP-CCYYMMDD  INP-DAYSADD  OUT-DAYSERR
                   OUT-YYERR      OUT-MMERR     OUT-DDERR
                   OUT-CCYYMMDD  OUT-FATALERR
```

Where:

INP-CCYYMMDD	X(08)	<INP>	= Input date mask [CCYYMMDD] [Must be 8-digits with expanded year];
INP-DAYSADD	9(05)	<INP>	= Unsigned number of days to add to input date (must always be 5 digit allocation);
OUT-DAYSERR	9(01)	<OUT>	= Number of days check results (0=OK >0=ERR);
OUT-YYERR	9(01)	<OUT>	= YY Validity Results (0=OK >0=ERR);
OUT-MMERR	9(01)	<OUT>	= MM Validity Results (0=OK >0=ERR);
OUT-DDERR	9(01)	<OUT>	= DD Validity Results (0=OK >0=ERR);
OUT-CCYYMMDD	X(08)	<OUT>	= Expanded date [CCYYMMDD].
OUT-FATALERR	9(05)	<OUT>	= Fatal Error Number -- 0 = No Errors: [4 = Any of above 4 error flags set > 0 13000 = Output CCYY reached 9999 14000 = Output MM > 12 but cannot increment year since output CCYY already reached 9999].

Detailed Functions:

1. First performs **date-validity check** on each component of the input date (checks actual calendar); If input date in error, returns individual error codes and does not perform addition functions.
2. Checks input "number of days to add" – discontinues processing if is unusable.

- Method is to add the number of days to the "day" value of the date, then adjust the result per the calendar. Allows for leap years.

Comments: If needed, call COBOL subroutine **DTEXP3** to expand incoming date from 6-digit (YYMMDD)-to-8-digits (CCYYMMDD) from calling program prior to this call.

See "copyproc" inserts "WS-DATES" (declarations) and "98-WINDOW" (procedures) for further details of how this Y2K expansion process works. See also the "Y2K Compliance COBOL Programmer's Reference Manual".

{End}

DTEXP3 — Subroutine {COBOL}

Purpose: Expand 6-digit date in yymmdd (year-month-day) format to 8-digits with 4-digit year.

Usage: Date expansion prior to use in date comparisons, calculations, display, and additional formatting, especially in terms of obtaining a **4-digit year** (for Y2K accuracy, etc.).

Calling Sequence:

```
CALL 'DTEXP3' USING  INP-YYMMDD    INP-MMCHK    INP-DDCHK
                    INP-ZEROS     INP-NINES    INP-SPACE
                    OUT-YYERR     OUT-MMERR     OUT-DDERR
                    OUT-CCYYMMDD
```

Where:

INP-YYMMDD	X(06)	<INP>	= Input date mask [YYMMDD];
INP-MMCHK	X(01)	<INP>	= Check option for mm N=No, Y=Yes;
INP-DDCHK	X(01)	<INP>	= Check option for dd N=No, Y=Yes (general non-calendar), C=Yes (per-calendar);
INP-ZEROS	X(04)	<INP>	= Allow all zeroes ZEOK = yes ZENO = no, treat as error;
INP-NINES	X(04)	<INP>	= Allow all nines NIOK = yes NINO = no, treat as error;
INP-SPACE	X(04)	<INP>	= Allow all spaces SPOK = yes SPNO = no, treat as error;
OUT-YYERR	9(01)	<OUT>	= YY Validity Results (0=OK >0=ERR);
OUT-MMERR	9(01)	<OUT>	= MM Validity Results (0=OK >0=ERR);
OUT-DDERR	9(01)	<OUT>	= DD Validity Results (0=OK >0=ERR);
OUT-CCYYMMDD	X(08)	<OUT>	= Expanded date [CCYYMMDD].

Detailed Functions:

1. First performs **date-validity check** on each component of the input date (checks actual calendar); If input date in error, returns individual error codes and does not perform addition functions.
2. Expands year (YY) to 4-digits (CCYY) – 00-49 expands to 2000-2049.

Comments: See “copyproc” inserts “WS-DATES” (declarations) and “98-WINDOW” (procedures) for further details of how this Y2K expansion process works.

See also the “Y2K Compliance COBOL Programmer’s Reference Manual”.

{End}

DTFM38 — Subroutine {COBOL}

Purpose: Generates **formatted** output for an 8-digit input date in 3 components [**CCYYMMDD**].

Output date formats supported:

1. "**MM_DD_CCYY**", where “_” is a variable delimiter;

Note -- Leading zeroes removed from month & day,

2. "**MONTH DD, CCYY**", where "**MONTH**" is the spelled-out month name;

Note -- Leading zero removed from day;

Resultant **formatted date** returned in an alpha character “mask” that can easily be transferred where needed.

Usage: Dates shown in reports, forms, or other types of “dated” text output.

Applications include leasing, mortgages, interest, investments, billings/collections, and any other financial applications; as well as mail or shipping documents.

Calling Sequence:

```
CALL 'DTFM38' USING  INP-CCYYMMDD  INP-DTFMT  INP-DELIM
                    OUT-YYERR    OUT-MMERR  OUT-DDERR
                    OUT-DTOUTPUT  OUT-NCHAR  OUT-FATALERR
```

Where:

INP-CCYYMMDD	X(08)	<INP>	=	Input date mask [CCYYMMDD] [Must be 8-digits with expanded year];
INP-DTFMT	X(01)	<INP>	=	Output date format (alpha 1-char) “1” = “ MM_DD_CCYY ” (“_” = INP-DELIM) “2” = “ MONTH DD, CCYY ”;
INP-DELIM	X(01)	<INP>	=	Delimiter for date output (format 1 <u>only</u>);
OUT-YYERR	9(01)	<OUT>	=	YY Validity Results (0=OK >0=ERR);
OUT-MMERR	9(01)	<OUT>	=	MM Validity Results (0=OK >0=ERR);
OUT-DDERR	9(01)	<OUT>	=	DD Validity Results (0=OK >0=ERR);

OUT-DTOUTPUT	X(18)	<OUT>	=	Formatted date output (text) mask [allocation must be X(18) always], with content packed left-justified ;
OUT-NCHAR	9(02)	<OUT>	=	Number of "date-value" characters filled into the date output mask (per above);
OUT-FATALERR	9(05)	<OUT>	=	Fatal Error Number -- 0 = No Errors: [4 = Any of above 3 error flags set > 0 5 = Incorrect INP-DTFMT value].

Detailed Functions:

1. First performs **date-validity check** on each component of the input date (checks actual calendar); If input date in error, returns individual error codes and does not perform addition functions.
2. Checks input "output date format number" – discontinues processing if incorrect.
3. If **date-format # 1**:
maps date into OUT-DTOUTPUT (left justified) in format "**MM_DD_CCYY**", where "**_**" is replaced with value of INP-DELIM, removing leading zeros in month, day;
Example: "20010409" into "4-9-2001" or "4/9/2001";
Example: "20011227" into "12-27-2001" or "12/27/2001".

If **date-format # 2**:
maps date into OUT-DTOUTPUT (left justified) in format "**MONTH DD, CCYY**", removing leading zero in day;
Example: "20010409" into "April 9, 2001";
Example: "20011227" into "December 27, 2001".
4. Output variable OUT-NCHAR is set to the number of date-format characters filled into OUT-DTOUTPUT – calling program can use this to pull "meaningful" text.

Comments: If needed, call COBOL subroutine **DTDEXP3** to expand incoming date from 6-digit (YYMMDD)-to-8-digits (CCYYMMDD) from calling program prior to this call.

See "copyproc" inserts "WS-DATES" (declarations) and "98-WINDOW" (procedures) for further details of how this Y2K expansion process works. See also the "Y2K Compliance COBOL Programmer's Reference Manual".

{End}

{End/ISO}