CA Calendar Routines®

Applications Manual

r6

Contents

Chapter 1: Overview

Chapter 2: Communications

Chapter 3: Function Codes

Appendix A: Return Codes

Appendix B: End Points

Index

Chapter 1: Overview

This chapter introduces you to the background, operation, standards and features of CA Calendar Routines.

This chapter includes the following sections:

- Background- Discusses the history and origin of CA Calendar Routines.
- Operation- Describes how your program communicates with CA Calendar Routines.
- Standards- Covers the conventions observed by CA Calendar Routines.
- Features- Describes some options that are available to the application programmer.

Background

CA Calendar Routines was written to address the following calendar-related problems with data processing systems:

- Data processing systems that divide by the two digit representation of years run the risk of terminating abnormally.
- Data processing systems that have leap-year sensitive logic should be aware of when it is a leap year. For example, a data processing system should be aware that the year 2008 is a leap year, but 2007 and 2009 are not.

CA Calendar Routines can handle any four digit year from 0001 through 9999. The product accomplishes this by using a full set of rules that can distinguish leap years from non-leap years.

The result is a set of conversion processes that accurately handle dates ranging from January 01, 0001 through December 31, 9999.

The Retail/Manufacturing Industry Calendar

The retail and manufacturing industries experience calendar-related problems that effect their planning periods and industry specific calendars. Most retail and manufacturing planning activity occurs on the order of days/ weeks/periods (i.e. months), rather than on the order of years like the financial and insurance industries.

The retail and manufacturing industry works on Special Fiscal Calendars that do not quite coincide with the familiar twelve-month calendar (on which CA Calendar Routines is currently designed). Some companies (particularly the retail and manufacturing industries) like to compare one quarter's sales or production to another as precisely as possible. However, traditional quarters (such as January-March, April-June, and so on) do not contain the exact same number of days.

Retail and manufacturing organizations artificially define their own calendar years as a number of quarters containing a specific number of weeks. The quarters normally contain either 13 or 14 weeks based on three periods (similar to months) consisting of four or five weeks. The fiscal quarters are aligned approximately with 12 or 13 periods (similar to months), consisting of 4-4-4 weeks, 4-4-5 weeks, 4-5-4 weeks, or 4-5-5 weeks. The beginning of each week is normally established by the organization based on a Friday or Saturday weekend.

Additionally, companies may need to shorten or lengthen the end of a retail/ manufacturing year, or they allow the days to accumulate and create an extra week every nth number of years. Because there is no industry specific standard, it is difficult to support the retail/manufacturing calendar algorithmically.

This release of CA Calendar Routines supports retail/manufacturing calendars as sets of dates pre-loaded into the product's fiscal calendar tables. This will allow you to answer frequently asked business questions such as "What was the beginning of the previous fiscal quarter?" or "What is the current fiscal week?"

Operation

This material describes how your program communicates with CA Calendar Routines. Communication is achieved through the Conversational Area. This is a special area that is referenced by your program and CA Calendar Routines.

The following text describes each step used when a Call or Link statement invokes CA Calendar Routines. See the illustration Operational Overview to Call/Link to CA Calendar Routines for more information.

1. Your program formats a request for CA Calendar Routines. To do this, you must specify the values of certain parameters in the Conversational Area.
2. Your program transfers control to CA Calendar Routines. The method of invocation can vary, but you probably will use a CALL statement or a LINK statement.
3. CA Calendar Routines receives control of execution.
4. CA Calendar Routines examines the request. If the request is valid, an answer will be formatted in the Conversational Area. If the request is not valid, an error number and text will be formatted instead.
5. CA Calendar Routines transfers control back to your program.
6. Your program receives control of execution.
7. Your program checks for an error condition. Your program must interrogate the return code before anything else because if the return code indicates a problem, then the other output areas have no meaning.
8. If the return code indicates no problem, then your program interprets the answer.

Operational Overview to Call/Link to CA Calendar Routines

Applications written in COBOL can CALL, LINK, or PERFORM CA Calendar Routines functions. Performing CA Calendar Routines functions is accomplished by performing in-line program source statements included in application programs. Several COPY member statements are required in the Working Storage and Procedure Division sections of the application program. The CA Calendar Routines functions that can be performed are limited to the following functions. See the chapter “Function Codes”, for more information:

Code	CA Calendar Routines Function
C101	DAYS BETWEEN
C102	DATE +/- N DAYS
C201	Get CALENDAR NEXT DAY

Code	CA Calendar Routines Function
C202	Get CALENDAR PREVIOUS DAY
U001	Get CURRENT DATE
U002	VALIDATE DATE
U003	Determine LEAP YEAR
U006	Get DAY OF WEEK
U007	DATE --> ABSOLUTE DAYS
U008	ABSOLUTE DAYS --> DATE
U009	REFORMATE DATE
U011	COMPARE Two DATES (same format)
U012	CONCATENATE CENTURY
U013	TRUNCATE CENTURY

Communication is achieved with the performed routines in the same manner as a CALL or LINK to the full-function CA Calendar Routines functions.

Standards

There are a variety of standards and conventions used within CA Calendar Routines. A few fundamental ones are described here.

- CA Calendar Routines treats Sunday as the first day of the week, and Saturday as the seventh day of the week. This corresponds to the arrangement of the days of the week on the calendar. Later you will be able to assign numerical values to the days-of-the-week, but the first position will still represent Sunday, and so on.
- CA Calendar Routines considers the number of days that have elapsed since January 01, 0001 as the ABSOLUTE DAYS. You will see this term used several times throughout this guide.
- CA Calendar Routines makes a distinction between a Calendar Day and a Processing Day.
  - A Calendar Day is any day of the calendar year.
  - A Processing Day is sometimes known as “business day” or “work day.” It is defined for each day of the week and may take into account holidays or other special dates such as period start and end dates.
- If you enter the literal CURRDATE into any FROM-DATE field or any TO-DATE field, the current date will be substituted for you.

- CA Calendar Routines maintains consistency between its two basic types of functions.
  - The first basic type of function computes the number of days between two dates. For example there are 3 days between March 15, 2007 and March 18, 2007.
  - The second basic type of function calculates a date based on a starting date and a numerical increment. For example, March 15, 2007 plus three days results in March 18, 2007.
- To maintain consistency, CA Calendar Routines enforces the following definitions.
  - FROM-DATE specifies the starting point.
  - TO-DATE specifies the ending point.
  - DIRECTION specifies the bearing. The DIRECTION is always from the FROM-DATE to the TO-DATE.
  - DISPLACEMENT specifies the actual number of physical days that lie between the two dates.

The principles of FROM-DATE, TO-DATE, DIRECTION, and DISPLACEMENT can be understood by studying these examples.

Days Between		**Date +/- *n* Days**	
From-Date	Mar 15, 2007	**From-Date**	Mar 15, 2007
To-Date	Mar 18, 2007	**Direc/Displ**	+3
Direc/Displ	+3	**To-Date**	Mar 18, 2007
From-Date	Mar 18, 2007	**From-Date**	Mar 18, 2007
To-Date	Mar 15, 2007	**Direc/Displ**	-3
Direc/Displ	-3	**To-Date**	Mar 15, 2007

Features

There are a variety of features and options available within CA Calendar Routines. A few basic ones are described here.

Multiple Holiday Tables.

CA Calendar Routines lets you choose between 1,296 different holiday tables (which you designate). This can be useful if you conduct business with other areas of commerce (banking, international, and so on.) or if certain areas within your organization use different holiday schedules.

Flexible Date Formats

CA Calendar Routines lets you specify dates in any of over 194 different numeric, alphanumeric, edited, or serial formats.

Free Form Definition of Processing Days

CA Calendar Routines lets you define each day of the week as a processing day or a non-processing day. When you define a processing day, you can take into consideration holidays or other special dates, such as period start and end dates.

Full Support of Fiscal Years, Quarters and Months

CA Calendar Routines lets you select any of the 12 months as the start of the fiscal year, and lets you specify day 01 through 28 as the start of the fiscal month.

Changeable End-Point Algorithms

CA Calendar Routines allows you to specify whether the FROM-DATE, the TO-DATE or both should be used in calculations.

Complete Support for Functions

CA Calendar Routines offers you over 300 functions. They fall into three basic groups.

- Utility functions, such as leap-year determination.
- Calendar Day functions, which consider all calendar days.
- Processing Day functions, which consider only processing days.

Descriptive Error Messages

CA Calendar Routines provides an eighty character error text with a three digit error number.

Integrated Support of Current Date-Time

CA Calendar Routines provides implicit and explicit access to the system's current date and time.

Special Fiscal Calendars

Two fiscal calendar definitions are included in CA Calendar Routines. These special calendars are used by retail and manufacturing organizations:

- 4-4-4 Fiscal Calendar Definition
- 4-5-4 Fiscal Calendar Definition

4-4-4 Fiscal Calendar Definition

The fiscal 4-4-4 Calendar consists of four-week periods (similar to a month), three and/or four period quarters, and a four-quarter Fiscal year.

- A week runs from Sunday through Saturday with days numbered 1 through 7.
- Weeks are numbered 1 through 52 / 53.
- The 4-4-4 Calendar year contains 13 periods of 52 weeks or 53 weeks (no partial weeks).
- The 4-4-4 fiscal Calendar is defined in standard calendar date notation (years, months, and days).
- Typically, every period contains four weeks. The exception is the last period of any year that contains 53 weeks. This last period contains five weeks.
- Each of the first three quarters of any year contains three periods (12 weeks). The last quarter contains four periods (16 or 17 weeks).
- The end of a year is determined by calculating the Saturday that is closest to 12/31 (December 31st) in that year:
  - When 12/31 falls on a Saturday, Sunday, Monday, or Tuesday, the Fiscal 4-4-4 year ends on the last Saturday of December of that year.
  - When 12/31 falls on Wednesday, Thursday, or Friday, the Fiscal 4-4-4 year ends on the first Saturday of January of the next year.

The following table contains examples of a 4-4-4 fiscal calendar definition.

Description	2007	2008	2009	2010
First Day of Year	12/31/2006	12/30/2007	01/04/2009	01/03/2010
Last Day of Year	12/29/2007	01/03/2009	01/02/2010	01/01/2011
Weeks in Year	52	53	52	52
Period 1 End	01/27/2007	01/26/2008	01/31/2009	01/30/2010
Period 2 End	02/24/2007	02/23/2008	02/28/2009	02/27/2010
Period 3 / Qtr 1 End	03/24/2007	03/22/2008	03/28/2009	03/27/2010
Period 4 End	04/21/2007	04/19/2008	04/25/2009	04/24/2010
Period 5 End	05/19/2007	05/17/2008	05/23/2009	05/22/2010
Period 6 / Qtr 2 End	06/16/2007	06/14/2008	06/20/2009	06/19/2010
Period 7 End	07/14/2007	07/12/2008	07/18/2009	07/17/2010
Period 8 End	08/11/2007	08/09/2008	08/15/2009	08/14/2010

Description	2007	2008	2009	2010
Period 9 / Qtr 3 End	09/08/2007	09/06/2008	09/12/2009	09/10/2010
Period 10 End	10/06/2007	10/04/2008	10/10/2009	10/09/2010
Period 11 End	11/03/2007	11/01/2008	11/07/2009	11/06/2010
Period 12 End	12/01/2007	11/29/2008	12/05/2009	12/04/2010
Period 13 / Qtr 4 End	12/29/2008	1/03/2009	01/02/2010	01/01/2011

4-5-4 Fiscal Calendar Definition

The fiscal 4-5-4 Calendar consists of four and five week periods (similar to a month), three period quarters, and a four-quarter Fiscal year.

- A week runs from Sunday through Saturday with days numbered 1 through 7.
- Weeks are numbered 1 through 52 / 53.
- The 4-5-4 Calendar year contains 12 periods of 52 weeks or 53 weeks (no partial weeks).
- The 4-5-4 fiscal Calendar is defined in standard calendar date notation (years, months, and days).
- Each quarter contains three periods. The first and third period contain four weeks. The second period of each quarter contains five weeks. (Hence the term 4-5-4.) The exception to this rule is when the last day of the year falls on January of the next calendar year. See the 4-5-4 fiscal calendar definition table for more information. When this occurs, a fifth week is added to the last period of the fourth quarter.
- The end of a year is determined by calculating the Saturday that is closest to 12/31 (December 31st) in that year:
  - When 12/31 falls on a Saturday, Sunday, Monday, or Tuesday, the Fiscal 4-5-4 year ends on the last Saturday of December of that year.
  - When 12/31 falls on Wednesday, Thursday, or Friday, the Fiscal 4-5-4 year ends on the first Saturday of January of the next year.

The following table contains examples of a 4-5-4 fiscal calendar definition.

Description	2007	2008	2009	2010
First Day of Year	12/31/2006	12/30/2007	01/04/2009	01/03/2010
Last Day of Year	12/29/2007	01/03/2009	01/02/2010	01/01/2011
Weeks in Year	52	53	52	52
Period 1 End	01/27/2007	01/26/2008	01/24/2009	01/29/2010
Period 2 End	03/03/2007	03/01/2008	02/28/2009	03/04/2010
Period 3 / Qtr 1 End	03/31/2007	03/29/2008	03/28/2009	04/01/2010
Period 4 End	04/28/2007	04/26/2008	04/25/2009	04/29/2010
Period 5 End	06/02/2007	05/31/2008	05/30/2009	06/03/2010
Period 6 / Qtr 2 End	06/30/2007	06/28/2008	06/27/2009	07/01/2010
Period 7 End	07/28/2007	07/26/2008	07/25/2009	07/29/2010
Period 8 End	09/01/2007	08/30/2008	08/29/2009	09/02/2010
Period 9 / Qtr 3 End	09/29/2007	09/27/2008	09/26/2009	09/30/2010
Period 10 End	11/03/2007	10/25/2008	10/24/2009	10/28/2010
Period 11 End	12/01/2007	11/29/2008	11/28/2009	12/02/2010
Period 12 End	12/29/2007	01/03/2009	01/02/2010	01/01/2011

Chapter 2: Communications

This chapter introduces you to the layout, business parameters, input parameters, output parameters, and extended IO areas of CA Calendar Routines.

This chapter includes the following sections:

- Layout- A brief description of the organization of the Conversation Area.
- Business Parameters- A brief description of the global settings.
- Input Parameters- A brief description of the information that you must provide to CA.
- Output Parameters- A brief description of the information that CA returns to you.
- Extended IO Areas- A brief description of the areas that are used to send and receive edited or serial date information.

Layout

The following example shows the Conversational Area, which contains three sections: Business Parameters, Input Parameters and Output Parameters. All of these entries will be described in their respective sections. (PLI is also provided).

```
01 TRC-CONVR-CONVERSATIONAL.
  05 TRC-CONVR-BUSINESS-PARAMETERS.
    10 TRC-CONVR-HOLIDAY-TBL PIC X(02) VALUE "01".
    10 TRC-CONVR-FISCAL-ID.
      15 TRC-CONVR-FISCYR-START PIC X(02) VALUE "01".
      15 TRC-CONVR-FISCMO-START PIC X(02) VALUE "01".
    10 TRC-CONVR-END-PNTS-DEF PIC X(01) VALUE "T".
    10 TRC-CONVR-DOW-STRING PIC X(07) VALUE "1234567".
    10 TRC-CONVR-PROC-DAY-DEF PIC X(07) VALUE "NEEEEEN".
    10 TRC-CONVR-CENTURY-BREAK PIC X(08) VALUE "19/20;68".
    10 TRC-CONVR-FROM-DATE-MASK PIC X(08) VALUE "YYMMDD--".
    10 TRC-CONVR-TO-DATE-MASK PIC X(08) VALUE "--YYMMDD".
    10 TRC-CONVR-OUT1-DATE-MASK PIC X(08) VALUE "YYMMDD--".
    10 TRC-CONVR-OUT2-DATE-MASK PIC X(08) VALUE "--YYMMDD".
    10 TRC-CONVR-OUT3-DATE-MASK PIC X(08) VALUE "-CCYYDDD".
    10 TRC-CONVR-VALIDATE-NULL PIC X(01) VALUE 'N'.
      88 TRC-CONVR-DO-NOT-EDIT-NULLS VALUE 'N'.
      88 TRC-CONVR-EDIT-NULLS VALUE 'Y'.
    10 TRC-CONVR-CASE PIC X(01) VALUE ' '.
      88 TRC-CONVR-UPPER-CASE VALUE 'U' OR ' '.
      88 TRC-CONVR-MIXED-CASE VALUE 'M'.
    10 FILLER PIC X(18) VALUE SPACE05   TRC-CONVR-INPUT-PARAMETERS.10 TRC-
CONVR-FUNCTION-CODE.
      15 TRC-CONVR-FUNCTION-TYPE   PIC X.
      15 TRC-CONVR-FUNCTION-NUMBER PIC X(3).
    10 TRC-CONVR-FROM-DATE-X   PIC X(8).
    10 TRC-CONVR-FROM-DATE-9 REDEFINES
TRC-CONVR-FROM-DATE-X   PIC 9(8).
```

```
   10 TRC-CONVR-TO-DATE-X    PIC X(8).
   10 TRC-CONVR-TO-DATE-9 REDEFINES TRC-CONVR-TO-DATE-X    PIC 9(8).
   10 TRC-CONVR-IN-NUMERIC-PARM    PIC S9(8) SIGN IS LEADING SEPARATE.
   10 TRC-CONVR-IN-NUM-PARM-REDEF REDEFINES TRC-CONVR-IN-NUMERIC-PARM.
     15 TRC-CONVR-IN-NUM-PARM-SIGN PIC X(01).
     15 TRC-CONVR-IN-NUM-PARM-VALUE PIC 9(08).
   10 TRC-CONVR-IN-ALPHA-PARM     PIC X(7).
   10 FILLER    PIC X(20) VALUE SPACE.
 05 TRC-CONVR-OUTPUT-PARAMETERS.
   10 TRC-CONVR-RETURN-CODE.
     15 TRC-CONVR-RETURN-NUM    PIC 9(3).
     88 TRC-CONVR-RETURN-GOOD VALUE 000.
     88 TRC-CONVR-RETURN-BAD  VALUE 001 THRU 999.
     15 TRC-CONVR-RETURN-TEXT    PIC X(80).
   10 TRC-CONVR-STDOUT-DATE-X     PIC X(8).
   10 TRC-CONVR-STDOUT-DATE-9 REDEFINES TRC-CONVR-STDOUT-DATE-X     PIC 9(8).
   10 TRC-CONVR-OUT1-DATE-X   PIC X(8).
   10 TRC-CONVR-OUT1-DATE-9 REDEFINES TRC-CONVR-OUT1-DATE-X   PIC 9(8).
   10 TRC-CONVR-OUT2-DATE-X   PIC X(8).
   10 TRC-CONVR-OUT2-DATE-9 REDEFINES TRC-CONVR-OUT2-DATE-X   PIC 9(8).
   10 TRC-CONVR-OUT3-DATE-X   PIC X(8).
   10 TRC-CONVR-OUT3-DATE-9 REDEFINES TRC-CONVR-OUT3-DATE-X   PIC 9(8).
   10 TRC-CONVR-OUT-NUMERIC-PARM   PIC S9(8) SIGN IS LEADING SEPARATE
   10 TRC-CONVR-OUT-NUM-PARM-REDEF REDEFINES TRC-CONVR-OUT-NUMERIC-PARM.
     15 TRC-CONVR-OUT-NUM-PARM-SIGN PIC X(01).
     15 TRC-CONVR-OUT-NUM-PARM-VALUE PIC 9(08)
   10 FILLER    PIC X(20) VALUE SPACE.
 05 TRC-CONVR-EXTENDED-IO-AREAS.
   10 TRC-CONVR-EDITED-IO.
     15 TRC-CONVR-FROM-DATE-E   PIC X(50).
     15 TRC-CONVR-TO-DATE-E    PIC X(50).
     15 TRC-CONVR-OUT1-DATE-E   PIC X(50).
     15 TRC-CONVR-OUT2-DATE-E   PIC X(50).
     15 TRC-CONVR-OUT3-DATE-E   PIC X(50).
   10 TRC-CONVR-SERIAL-IO REDEFINES TRC-CONVR-EDITED-IO.
     15 TRC-CONVR-FROM-DATE-S   PIC S9(8) SIGN IS LEADING SEPARATE.
     15 FILLER    PIC X(41).
     15 TRC-CONVR-TO-DATE-S    PIC S9(8) SIGN IS LEADING SEPARATE.
     15 FILLER    PIC X(41).
     15 TRC-CONVR-OUT1-DATE-S   PIC S9(8) SIGN IS LEADING SEPARATE.
     15 FILLER    PIC X(41).
     15 TRC-CONVR-OUT2-DATE-S   PIC S9(8) SIGN IS LEADING SEPARATE.
     15 FILLER    PIC X(41).
     15 TRC-CONVR-OUT3-DATE-S   PIC S9(8) SIGN IS LEADING SEPARATE.
     15 FILLER    PIC X(41).
 05 FILLER  PIC X(      151) VALUE SPACE.
 05 TRC-CONVR-DB2-TABLE-ID
   10 TRC-CONVR-CICS-HOLIDAY-VSAM   PIC X(08) VALUE 'TRCHLF01'.
   10 FILLER REDEFINES TRC-CONVR-CICS-HOLIDAY-VSAM.
     15 FILLER    PIC X(06).
     15 TRC-CONVR-CICS-HOLIDAY-ID PIC X(02).
   10 TRC-DDNAME  PIC X(52) VALUE SPACE.
 05 TRC-CONVR-RESERVED-FOR-USER    PIC X(      150) VALUE SPACE.
 05 TRC-CONVR-CA-ONLY
   10 FILLER   PIC X(79) VALUE SPACE.
   10 TRC-CONVR-CUSTOMER-ID  PIC X(    10) VALUE SPACE.
   10 TRC-CONVR-VERSION-ID   PIC X(04) VALUE SPACE.
   10 TRC-CONVR-SELF-CHECK   PIC X(07) VALUE SPACE.
```

Business Parameters

The Business Parameters are global settings that are in effect for every invocation of CA Calendar Routines. Your System Administrator has assigned initial values to serve the needs of most of your company. Therefore, you can probably leave all the values untouched. However, if you have a special requirement, you may change the value of any parameter at any time.

There are twelve Business Parameters in the Conversational Area. The example Conversational Area in the Layout section showed you all of the Business Parameters together. The following text describes each parameter individually.

TRC-CONVR-HOLIDAY-TBL

This parameter specifies the holiday table that you would like to reference. You can choose any table from 00 through ZZ. Your System Administrator can tell you which tables contain the holidays (or special dates like period start/end dates) that you would like to reference.

TRC-CONVR-FISCAL-ID (Special Fiscal Calendar Functions Only)

This parameter is a combination of the TRC-CONVR-FISCYR-START and TRC-CONVR-FISCMO-START parameters (fields). Use the TRC-CONVR-FISCAL-ID for all special fiscal calendar functions (F401, F402, C/P 403, and C/P 404). The parameter specifies which Fiscal Calendar table to use when processing Fiscal Calendar functions. Values are $444, $454, $445, and $455.

TRC-CONVR-FISCYR-START

This parameter specifies the first month of the fiscal year. Its value ranges from 01 through 12. CA uses this value when it performs a function related to the fiscal year, quarter, or month. Typical questions include: “What is the beginning of the next fiscal quarter?” or “What was the end of the previous fiscal year?” If your company's fiscal year coincides with the calendar year, then this parameter should be set to 01.

TRC-CONVR-FISCMO-START

This parameter specifies the first day of the fiscal month. Its value can range from 01 through 28. CA Calendar Routines uses this value when it performs a function related to the fiscal year, quarter, or month. Typical questions include: “What is the beginning of the current fiscal month?” or “What was the end of the previous fiscal quarter?” If your company's fiscal month coincides with the calendar month, then this parameter should be set to the value of 01.

TRC-CONVR-END-PNTS-DEF

This parameter specifies which of the two end points CA Calendar Routines should consider when performing calculations. There are four possible values for this business parameter:

- B = BOTH (FROM-DATE and TO-DATE)
- N = NEITHER (neither one)
- F = FROM-DATE, only (just the FROM-DATE)
- T = TO-DATE, only (just the TO-DATE)

We highly recommend using T to yield common sense answers. Consider a simple example. Suppose that we want to know how many days lie between March 15 and March 18.

There are four possibilities depending on the value of END-PNTS-DEF:

If B is specified, the answer is 4 (i.e., 15th, 16th, 17th, 18th)

If N is specified, the answer is 2 (i.e., 16th, 17th)

If F is specified, the answer is 3 (i.e., 15th, 16th, 17th)

If T is specified, the answer is 3 (i.e., 16th, 17th, 18th

The TRC-CONVR-END-PNTS-DEF parameter is used explicitly by the following functions:

- 100 series
- 200 series except 207

  Note: This parameter is set implicitly to T for function 207.
- 300 series
- 500 series
- 600 series
- 700 series
- 800 series

  Note: This parameter is set implicitly to B for the 800 series.

When you consider the value of this parameter, keep the following considerations in mind.

- If you use the value B or N and you specify dates that are equal, or a direction/displacement of zero, then you may receive an error message. This error message will say that you attempted a restricted operation. See the appendices “Return Codes” and “End Points” for more information.

  CA restricts certain operations. Consider a request to calculate the number of days between January 14, 2007 and January 14, 2007; and neither end points is to be included. The solution to such a problem is undefined; and CA has decided to restrict such operations, rather than to assign an arbitrary answer, such as zero.

- It is important to understand the manner in which processing day and end point work together. As an example, let us suppose that you want to know the number of processing days from September 06, 2007 to September 09, 2007. Let us further suppose that September 06, 2007 is a holiday and that you do not want to consider it a processing day. Under these circumstances, you can see the importance of specifying the end points that should be included or excluded.

  – If the end points parameter is set to T (TO-DATE, only), then the answer is 3. (The range is 07, 08, 09; and all days qualify.)

  – If the end points parameter is set to F (FROM-DATE, only), then the answer is 2. (The range is 06, 07, 08; but 06 does not qualify because it was not a processing day.)

TRC-CONVR-DOW-STRING

This parameter lets you define the numerical values that correspond to each day of the week. The parameter is composed of seven different settings; one for each day of the week. Following the CA standard, position 1 corresponds to Sunday and position 7 corresponds to Saturday. Any numerical value is accepted, although repeat values are not allowed. There are three sequences that you might like to consider for your application.

CA Standard	1,2,3,4,5,6,7
IBM CICS	0,1,2,3,4,5,6
ANSI (and IBM COBOL II)	7,1,2,3,4,5,6

TRC-CONVR-PROC-DAY-DEF

This parameter is composed of seven different settings; one for each day of the week. Following the CA standard, position number 1 corresponds to Sunday and position number 7 corresponds to Saturday. For each day, you must specify one of the values shown next. Each value specifies a different rule regarding the inclusion or exclusion of holidays. Their meanings are:

A = Always (Always a processing day)

N = Never (Never a processing day)

E = Exclude (A processing day only if not a Holiday)

I = IncludeF (A processing day only if a Holiday)

As an example, if your site had the value NEEEEEN, it would mean that your site would not process on weekends and that your site would not process weekday holidays.

TRC-CONVR-CENTURY-BREAK

This parameter lets you tell CA how to assign a value to the century when the supplied date does not contain the century itself (for example, YYMMDD).

Note: The previous format AA/BB;CC such as 19/20;68 is still supported.

With this parameter, the rule is read as follows: If the year is 00 through 67, then the century is assigned the value 20, otherwise if the year is 68 through 99, then the century is assigned the value 19.

The new syntax is SLIDE-nn. For example SLIDE-20.

In the example AA equals the current year minus 20 years and BB equals the current year plus 79 years, the rule is read as follows: Let the window extend from AA through BB where:

- AA = current-year – 20
- BB = current-year + 79

For example, for 1997 the window would range from 1977 through 2076, and in 1998, the window would automatically advance from 1978 through 2077.

TRC-CONVR-FROM-DATE-MASK

This parameter lets you define the format of the input FROM-DATE. There are several categories of masks.

Note: The proper input area (such as -9, -X, -E, or -S) depending on the type of mask.

Using Date Elements

You can describe the arrangement of various pieces of date information, such as year, month, and day. There are five types of elements: CC century, YY year, MM Gregorian Month, DD Gregorian Day, and DDD Julian Day. The character - corresponds to leading zeroes (numeric) or trailing spaces (alphanumeric). (Use FROM-DATE-X or FROM-DATE-9 as the input area.)

```
"---YYDDD" Numeric Move of YYDDD
"YYMMDD--" Alphanumeric Move of YYMMDD
```

Single Century Designation

You can also use the following masks that will let you designate a century as a single character.

- If H is 8, the century is 18.
- If H is 9, the century is 19.
- If H is 0, the century is 20.
- If R is 0 the century is 19.
- If “R” is 1, the century is 20.
- If S is 1 the century is 19.
- If S is 2, the century is 20.
- If T is 0, the century is 18.
- If T is 1, the century is 19.
- If T is 2, the century is 20.

The character - corresponds to leading zeroes (numeric) or trailing spaces (alphanumeric). (Use FROM-DATE-X or FROM-DATE-9 as the input area.)

```
"-HYYMMDD" "-HMMDDYY" "HYYMMDD-" "HMMDDYY-"
"--HYYDDD" "--HDDDYY" "HYYDDD--" "HDDDYY--"
"-RYYMMDD" "-RMMDDYY" "RYYMMDD-" "RMMDDYY-"
"--RYYDDD" "--RDDDYY" "RYYDDD--" "RDDDYY--"
"-SYYMMDD" "-SMMDDYY" "SYYMMDD-" "SMMDDYY-"
"--SYYDDD" "--SDDDYY" "SYYDDD--" "SDDDYY--"
"-TYYMMDD" "-TMMDDYY" "TYYMMDD-" "TMMDDYY-"
"--TYYDDD" "--TDDDYY" "TYYDDD--" "TDDDYY--"
```

360-Day Calendar Designation

You can also use the following masks to designate days within a 360-DAY calendar year. These masks are only allowed in the K300 series of functions, and KKK represents a value from 001 through 360. The character “-” corresponds to leading zeroes (numeric) or trailing spaces (alphanumeric). (Ue FROM-DATE-X or FROM-DATE-9 as the input area.)

```
"CCYYKKK-" "-CCYYKKK" "KKKCCYY-" "-KKKCCYY"
"YYKKK---" "---YYKKK" "KKKYY---" "---KKKYY"
```

Edited Masks

You can also use the following edited masks. (Use FROM-DATE-E as the input area.)

Normal Edited

```
"E-M/D/Y " equals MM/DD/YY
"E-Y/M/D " equals YY/MM/DD
"E-Y-M-D " equals YY-MM-DD
"E-M-D-Y " equals MM-DD-YY
"E-D-M-Y " equals DD-MM-YY
"E-CY-M-D" equals CCYY-MM-DD
"E-M-D-CY" equals MM-DD-CCYY
"E-D-M-CY" equals DD-MM-CCYY
"E-YY-DDD" equals YY-DDD
"E-M D Y " equals MM DD YY
"E-M D CY" equals MM DD CCYY
"E-DMONY " equals DDmmmYY e.g. 20FEB95
"E-DMONCY" equals DDmmmCCYY e.g. 20FEB1995
```

Note: "E-DMONY" and "E-DMONCY" are also included as "E-TDS-NN" formats. This was done to support the non-English versions of these date masks. These date masks will continue to be supported in the English language versions.

Special DB2 Formats

```
"E-DB2ISO" equals CCYY-MM-DD
"E-DB2USA" equals MM/DD/CCYY
"E-DB2EUR" equals DD.MM.CCYY
"E-DB2JIS" equals CCYY-MM-DD
```

Special AS400 Formats

```
"E-A4*ISO" equals CCYY-MM-DD
"E-A4*USA" equals MM/DD/CCYY
"E-A4*EUR" equals DD.MM.CCYY
"E-A4*JIS" equals CCYY-MM-DD
"E-A4*MDY" equals MM/DD/YY
"E-A4*DMY" equals DD/MM/YY
"E-A4*YMD" equals YY/MM/DD
"E-A4*JUL" equals YY/DDD
```

User Requested

```
"E-TDS-01"  equals DD mmm YY (18 DEC 95)
"E-TDS-02"  equals DD-mmm-YY (18-DEC-95)
"E-TDS-03"  equals DD/mmm/YY (18/DEC/95)
"E-TDS-04"  equals DD mmm CCYY  (18 DEC 1995)
"E-TDS-05"  equals DD-mmm-CCYY  (18-DEC-1995)
"E-TDS-06"  equals DD/mmm/CCYY  (18/DEC/1995)
"E-TDS-07"  equals mmm. DD, CCYY (DEC. 18, 1995)
"E-TDS-08"  equals mmm DD CCYY  (DEC 18 1995)
```

```
"E-TDS-09"  equals mmm-DD-CCYY  (DEC-18-1995)
"E-TDS-10"  equals mmm/DD/CCYY  (DEC/18/1995)
"E-TDS-11"  equals MMMMM DD, CCYY  (DECEMBER 18, 1995)
"E-TDS-12"  equals DD.MM.YY (18.12.95)
"E-TDS-13"  equals CCYY MMM DD  (1995 DEC 18)
"E-TDS-14"  equals CCYY/MM/DD  (1995/12/18)
"E-TDS-15"  equals MM/DD/YYCC  (18/12/9519)
"E-TDS-16"  equals DDmmmYY  (18DEC95)
"E-TDS-17"  equals DDmmmCCYY (18DEC1995)
"E-TDS-18"  equals ddd DD mmm YY (MON 18 DEC 95)
"E-TDS-19"  equals ddd DD-mmm-YY (MON 18-DEC-95)
"E-TDS-20"  equals ddd DD/mmm/YY (MON 18/DEC/95)
"E-TDS-21"  equals ddd DD mmm CCYY (MON 18 DEC 1995)
"E-TDS-22"  equals ddd DD-mmm-CCYY (MON 18-DEC-1995)
"E-TDS-23"  equals ddd DD/mmm/CCYY (MON 18/DEC/1995)
"E-TDS-24"  equals ddd mmm. DD. CCYY  (MON DEC. 18. 1995)
"E-TDS-25"  equals ddd mmm DD CCYY (MON DEC 18 1995)
"E-TDS-26"  equals ddd mmm-DD-CCYY (MON DEC-18-1995)
"E-TDS-27"  equals ddd mmm/DD/CCYY (MON DEC/18/1995)
"E-TDS-28"  equals ddd MMMMM DD, CCYY (MON DECEMBER 18, 1995)
"E-TDS-29"  equals ddd DD.MM.YY (MON 18.12.95)
"E-TDS-30"  equals ddd CCYY mmm DD (MON 1995 DEC 18)
"E-TDS-31"  equals ddd CCYY/MM/DD  (MON 1995/12/18)
"E-TDS-32"  equals ddd MM/DD/YYCC  (MON 12/18/9519)
"E-TDS-33"  equals ddd DDmmmYY  (MON 18DEC95)
"E-TDS-34"  equals ddd DDmmmCCYY (MON 18DEC1995)
"E-TDS-35" equals DDDDD DD mmm YY (MONDAY 18 DEC 95)
"E-TDS-36"  equals DDDDD DD-mmm-YY (MONDAY 18-DEC-95)
"E-TDS-37"  equals DDDDD DD/mmm/YY (MONDAY 18/DEC/95)
"E-TDS-38" equals DDDDD DD mmm CCYY (MONDAY 18 DEC 1995)
"E-TDS-39"  equals DDDDD DD-mmm-CCYY  (MONDAY 18-DEC-1995)
"E-TDS-40"  equals DDDDD DD/mmm/CCYY  (MONDAY 18/DEC/1995)
"E-TDS-41"  equals DDDDD mmm. DD. CCYY (MONDAY DEC. 18. 1995)
"E-TDS-42"  equals DDDDD mmm DD CCYY  (MONDAY DEC 18 1995)
"E-TDS-43"  equals DDDDD mmm-DD-CCYY  (MONDAY DEC-18-1995)
"E-TDS-44"  equals DDDDD mmm/DD/CCYY  (MONDAY DEC/18/1995)
"E-TDS-45"  equals DDDDD MMMMM DD, CCYY (MONDAY DECEMBER 18,1995)
"E-TDS-46"  equals DDDDD mmm DD.MM.YY (MONDAY 18.12.95)
"E-TDS-47"  equals DDDDD CCYY mmm DD (MONDAY 1995 DEC 18)
"E-TDS-48"  equals DDDDD CCYY/MM/DD (MONDAY 1995/12/18)
"E-TDS-49"  equals DDDDD MM/DD/YYCC (MONDAY 12/18/9519)
"E-TDS-50"  equals DDDDD DDmmmYY (MONDAY 18DEC95)
"E-TDS-51"  equals DDDDD DDmmmCCYY (MONDAY 18DEC1995)
"E-TDS-52"  equals MMMMM CCYY  (DECEMBER 1995)
"E-TDS-53"  equals mmm DD, CCYY (DEC 18, 1995)
"E-TDS-54"  equals YY.DD (99.351)
"E-TDS-55"  equals CCYY.DDD (1999.351)
"E-TDS-56"  equals mmm.DD, CCYY (DEC.18, 1995)
"E-TDS-57"  equals ZM-ZD-ZY (2-18-95)
"E-TDS-58"  equals ZM/ZD/ZY (2/18/95)
```

Special Null Mask

Use Special Null Masks in Output Fields if you do not want multiple output date formats. This can improve execution performance.

"--------" equals null mask (i.e. no output produced)

User Requested – Non-English Date Masks

Date masks identical in format to English E-TDS-nn date masks are provided in the following languages:

IBM- Country Code	Country
AT	Australia
DK	Denmark
FI	Finland
FR	France
NL	Netherlands
IT	Italy
DE	Germany
NO	Norway
SE	Sweden
ES	Spain

To obtain a non-English User Requested date format, modify the Edited Date mask as follows:

```
E-TDS-nn ETDSnnxx xx = above country code
E-TDS-45 (English) ETDS45ES (Spanish)
MONDAY DECEMBER 18, 2007 LUNES DICIEMBRE 18, 2007
```

Special Null Mask

Use Special Null Masks in Output Fields if you do not want multiple output date formats. This can improve execution performance.

"--------" equals null mask (i.e. no output produced)

9-S Complement Masks

You can also use the following 9-S complement masks. These dates represent the original date for which each digit has been subtracted from 9. (Use FROM-DATE-X or FROM-DATE-9 as the input area.)

```
"9-CYMD " equals 9's compl of CCYYMMDD
"9-N-YMD " equals 9's compl of YYMMDD PIC 9
"9-N-MDY " equals 9's compl of MMDDYY PIC 9
"9-N-CYJ " equals 9's compl of CCYYDDD PIC 9
"9-N-YJ " equals 9's compl of YYDDD PIC 9
```

```
"9-A-YMD " equals 9's compl of YYMMDD PIC X
"9-A-MDY " equals 9's compl of MMDDYY PIC X
"9-A-CYJ " equals 9's compl of CCYYDDD PIC X
"9-A-YJ " equals 9's compl of YYDDD PIC X
```

SERIAL-DATE Masks

You can also use the following SERIAL-DATE masks. These dates represent a displacement from a predefined reference date. Several such conventions exist for you to choose from. TDS is the CA Standard, S-1900G follows the 100 leap year rule, S-1900NG does not. (Use FROM-DATE-S as the input area.)

```
Convention Low High Reference DayCount
S-TDS 1 3652056 Jan 01, 0001 1
S-DB2 1 3652059 Jan 01, 0001 1
S-LE370 -577734 3074324 Oct 15, 1582 1
S-COB370 -584387 3067671 Jan 01, 1601 1
S-ANSI89 -584387 3067671 Jan 01, 1601 1
S-CICS-A -693594 2958464 Jan 01, 1900 1
S-1900G -693594 2958464 Jan 01, 1900 1
S-1900NG -693594 2958468 Jan 01, 1900 1
```

User defined SERIAL DATE masks (Snnnnnnn) let you define the low and high references shown previously. The format for this date mask is shown in the following example:

```
Snnnnnnn User serial date based on date value supplied in “nnnnnnn”.
Digits 6-7 = Day Value (e.g., 15)
Digits 4-5 = Month Value (e.g., 12)
Digits 2-3 = Year Value (e.g., 97)
Digit 1 designates the century number, where:0 for 2000 5 for 15001 for 2100 6
for 16002 for 2200 7 for 17003 for 2300 8 for 18004 for 2400 9 for 1900
Serial date mask identifier (constant)
```

SPECIAL FISCAL CALENDAR Date Masks

Fiscal Calendar date information is described by five elements:

- “CC” Century
- “YY” Year
- “PP” Period (similar to a month)
- “WW” Week
- “DD” Day of Period

The character “-” corresponds to leading zeroes (numeric) or trailing spaces (alphanumeric). The applications programmer must use FROM-DATE-X or FROM-DATE-9 as the input area.

- --YYPPDD Numeric Move of YYPPDD
- ----YYWW Numeric YYWW
- YYPPDD--Alphanumeric Move of YYMMDD

- CCYYWW--Alphanumeric CCYYWW

The Alphanumeric Fiscal Calendar date masks are:

- YY------YEAR (century omitted)
- CCYY----YEAR
- YYQQ----YEAR (century omitted)/Quarter
- CCYYQQ--YEAR/Quarter
- YYPP----YEAR (century omitted)/Period
- CCYYPP--YEAR/Period
- YYPPDD--YEAR (century omitted)/Period/Day-of-period
- CCYYPPDD YEAR/Period/Day-of-period
- YYWW----YEAR (century omitted)/Week number
- CCYYWW--YEAR/Week number

The Numeric Fiscal Calendar date masks are:

- ------YY YEAR (century omitted)
- ----CCYY YEAR
- ----YYQQ YEAR (century omitted)/Quarter
- --CCYYQQ YEAR/Quarter
- ----YYPP YEAR (century omitted)/Period
- --CCYYPP YEAR/Period
- --YYPPDD YEAR (century omitted)/Period/Day-of-period
- ----YYWW YEAR (century omitted/Week number
- --CCYYWW YEAR/Week number

TRC-CONVR-TO-DATE-MASK

This parameter defines the format of TO-DATE. See TRC-CONVR-FROM-DATE-MASK for more information.

TRC-CONVR-OUT1-DATE-MASK

This parameter defines the format of OUT1-DATE. See TRC-CONVR-FROM-DATE-MASK for more information.

TRC-CONVR-OUT2-DATE-MASK

This parameter defines the format of OUT2-DATE. See TRC-CONVR-FROM-DATE-MASK for more information. Pre-set this parameter to the null mask for efficiency.

TRC-CONVR-OUT3-DATE-MASK

This parameter defines the format of OUT3-DATE. See TRC-CONVR-FROM-DATE-MASK for more information. Pre-set this paramater to the null mask for efficiency.

TRC-CONVR-VALIDATE-NULL

This parameter defines the extent of input date validation performed by the Calendar Routines (see TRC-CONVR-FROM-DATE and TRC-CONVR-TO-DATE for more information) on Standard Date Element (see TRC-CONVR-FROM-DATE-MASK, Using Date Elements for more information).

The default setting for this version of CA Calendar Routines is with the Validate-Null switch set to Y. Y designates full validation of all date input positions in the eight-character FROM and TO dates. N designates no validation of non-date element positions.

```
Input VALIDATE Input Validation
DATE MASK NULL Field Condition
YYMMDD-N 971231@@ No errors
YYMMDD-Y 971231@@ Error
---YYDDD N bbb97121 No errors
---YYDDD Y bbb97121 Error
```

Setting this switch to N can be useful when the user does not want to validate non date elements for conversion purposes.

TRC-CONVR-CASE

This parameter determines the case setting for alphabetic date output literals. The default setting is U or blank. The default setting instructs the CA Calendar Routines to output all alphabetic literals in upper case (e.g. JANUARY 1, 2007). Change this value to M if you want the Calendar Routines to output date literals in mixed case (e.g. Monday January 1, 2007).

Input Parameters

The Input parameters are the values that you pass to CA. There are five Input parameters in the Conversational Area. The Conversational Area example in the Layout section showed you all of the Input Parameters together. The following text describes each parameter individually.

TRC-CONVR-FUNCTION-CODE

This specifies the task that you want CA to perform. It contains the following components:

- Function Type, which is specified by a one-character code.
  - U Utility Function
  - C Calendar Day Function
  - P Processing Day Function
  - K Calculations on the 360-Day Calendar
  - F Fiscal Calendar Function
- Function Number, which is specified by a three-digit value.

For a description of Function Type and Function Number, see the chapter “Function Codes”.

TRC-CONVR-FROM-DATE

This parameter serves two purposes. If a function requires only one date, then it should go into the FROM-DATE field. Otherwise, if a function requires a FROM-DATE and TO-DATE, then the FROM-DATE goes into the FROM-DATE field. The format of the FROM-DATE is defined by the FROM-DATE-MASK:

FROM-DATE-9	Numeric Dates
FROM-DATE-X	Alphanumeric Dates
FROM-DATE-E	Edited Dates
FROM-DATE-S	Serial Dates

Also, if you enter the literal CURRDATE into the FROM-DATE field, the current date will be substituted for you.

TRC-CONVR-TO-DATE

This parameter specifies the TO-DATE for functions that require such a date. The format of the TO-DATE is defined by the TO-DATE-MASK.

TO-DATE-9	Numeric Dates
TO-DATE-X	Alphanumeric Dates
TO-DATE-E	Edited Dates
TO-DATE-S	Serial Dates such as PIC 9 (whether DISPLAY, BINARY, PACKED)

Also, if you enter the literal CURRDATE into the TO-DATE field, the current date will be substituted for you.

TRC-CONVR-IN-NUMERIC-PARM

This parameter specifies the DIRECTION and DISPLACEMENT in number of days, weeks, months, fiscal Calendar periods, or years (depending on the function). For example, if you wanted to know the date that is +4 days from March 15, 1997, then you would put a value of +4 into IN-NUMERIC-PARM.

For function 207, IN-NUMERIC-PARM must be non-zero. For odd functions in the range 801 through 836, the value of IN-NUMERIC-PARM must be greater than zero. For even functions in the range 801 through 836, the value of IN-NUMERIC-PARM must be less than zero.

Note: For compilers that do not recognize SIGN IS LEADING SEPARATE, CA offers the following redefinition:

```
10 TRC-CONVR-IN-NUMERIC-PARM-ADSO.
15 TRC-CONVR-IN-NUM-PARM-SIGN  PIC X(01).
15 TRC-CONVR-IN-NUM-PARM-VALUE  PIC 9(08).
```

TRC-CONVR-IN-ALPHA-PARM

This parameter has different meanings for different functions.

- Specifies the days of the week that you want to consider when working with Function 207 and the Functions 801-836. The parameter actually consists of seven different settings, which correspond to Sunday through Saturday. You must specify Y for yes or N for no, so that CA will know which days of the week to include in calculations.
- Specifies the duration of WEEKS, M-ENF, M-NOENF, Y-ENF, Y-NOENF for Function 103.
- Specifies the duration of K360-DY, K360-MO, or K360-YR for Function 304.

- Specifies the duration of ENFORCE or NO-ENF for Function U010.
- Specifies the following for functions F401, F402, C/P 404:

Parameter	Explanation F401 Calendar Special Date to Fiscal Date
F-BOY	beginning of Fiscal Year
F-BOQ	beginning of Fiscal Quarter
F-BOP	beginning of Fiscal Period
F-BOW	beginning of Fiscal Week
F-EOY	end of Fiscal Year
F-EOQ	end of Fiscal Quarter
F-EOP	end of Fiscal Period
F-EOW	end of Fiscal Week

Parameter	Explanation F402 Special Fiscal Date to Calendar Date
F-BEGIN	Beginning date
F-END	Ending date

Parameter	Explanation C/P 404 Special Fiscal Date + / - *n* Days, Weeks, Periods, Quarters or Years
F-DAYS	increment/decrement n days
F-WEEKS	increment/decrement n weeks
F-PER	increment/decrement n periods
F-QTR	increment/decrement n quarters
F-YEARS	increment/decrement n years

Output Parameters

The Output parameters are the values that you receive from CA. There are six Output Parameters in the Conversational Area. The Conversational Area example in the Layout section showed you all of the Output Parameters together. The following text describes each parameter individually.

TRC-CONVR-RETURN-CODE

This parameter contains information regarding the completion of the task. It contains two components. See the appendix “Return Codes” for a description of all of the Return Codes in greater detail.

Return Number, which specifies a return with no problems (000) or a return with problems (001-999). Your program should test the value of the Return Number every time you invoke CA. Remember, if the return number specifies a return with problems, then the other output areas have no meaning.

Return Text, which elaborates on the meaning of the Return Number. If the Return Number is 000, then the Return Text will say GOOD RETURN. If the Return Number is 001-999, then the Return Text will contain a descriptive message.

TRC-CONVR-STDOUT-DATE

If the function that you have specified results in the calculation of a date, then that date will be found in four different date fields. The first of the output date fields is referred to as STDOUT-DATE. It represents the CA standard of CCYYMMDD. STDOUT-DATE is provided for you automatically.

TRC-CONVR-OUT1-DATE-X (-9)

This is the second output date area. The exact format is specified by OUT1-DATE-MASK.

Note: If OUT1-DATE-MASK denotes an edited or serial date, the answer will be in OUT1-DATE-E or OUT1-DATE-S respectively.

TRC-CONVR-OUT2-DATE-X (-9)

This is the third output date area. The exact format is specified by OUT2-DATE-MASK.

Note: If OUT2-DATE-MASK denotes an edited or serial date, the answer will be in OUT2-DATE-E or OUT2-DATE-S respectively.

TRC-CONVR-OUT3-DATE-X (-9)

This is the fourth output date area. The exact format is specified by OUT3-DATE-MASK.

Note: If OUT3-DATE-MASK denotes an edited or serial date, the answer will be in OUT3-DATE-E or OUT3-DATE-S respectively.

TRC-CONVR-OUT-NUMERIC-PARM

This parameter serves two main purposes.

- Most of the time, OUT-NUMERIC-PARM specifies the answer to such questions as “How many days are there from August 15, 2007 to August 23, 2007?” In this example the answer is +8. For inquiries of this type, the answer always specifies Direction and Displacement.
- The rest of the time, OUT-NUMERIC-PARM denotes answers to specific questions posed by Utility Functions U002 through U007. These situations are described in the following table.

FUNC	TASK	OUT NUMERIC-PARM
002	Valid Date Y/N	0-NO; 1-YES
003	Leap Year Y/N	0-NO; 1-YES
004	Holiday Y/N	0-NO; 1-YES
005	Processing Day Y/N	0-NO; 1-YES
006	Day of the Week	See DOW-STRING for the numerical values that correspond to Sunday through Saturday
007	Absolute Days	000000001-03652056: (000000001=Jan 01, 0001; (03652056= Dec 31, 9999)
011	Compare (From/To Dates)	0; From date EQUAL to date +1; From date GREATER THAN to date -1; From date LESS THAN to date

- For Fiscal Calendar Functions where a Fiscal Calendar date is created, the TRC-CONVR-OUT-NUM-PARM-VALUE contains the format QQPPWWDD:
  - QQ Fiscal Quarter number (01-04)
  - PP Fiscal Period number (01-12/13)
  - WW Fiscal Week number (01-52/53)
  - DD Fiscal Day-of-Period number (01-37)

  Note: For compilers that do not recognize SIGN IS LEADING SEPARATE, CA offers the following redefinition:

```
10 TRC-CONVR-OUT-NUMERIC-PARM-ADSO.
15 TRC-CONVR-OUT-NUM-PARM-SIGN PIC X(01).
15 TRC-CONVR-OUT-NUM-PARM-VALUE PIC 9(08).
```

Extended-IO-Areas

Extended IO areas are used to send or receive edited dates and serial dates. There is one area for each date such as FROM, TO, OUT1, OUT2, and OUT3. Use the -E or -S area as the corresponding mask warrants.

Edited Dates

TRC-CONVR-FROM-DATE-E

Used if the FROM-DATE-MASK is one of the edited types, such as E-M/D/Y.

TRC-CONVR-TO-DATE-E

Used if the TO-DATE-MASK is one of the edited types, such as, E-M/D/Y.

TRC-CONVR-OUT1-DATE-E

Used if the OUT1-DATE-MASK is one of the edited types, such as, E-M/D/Y.

TRC-CONVR-OUT2-DATE-E

Used if the OUT2-DATE-MASK is one of the edited types, such as, E-M/D/Y.

TRC-CONVR-OUT3-DATE-E

Used if the OUT3-DATE-MASK is one of the edited types, such as, E-M/D/Y.

Serial Dates

TRC-CONVR-FROM-DATE-S

Used if the FROM-DATE-MASK is one of the serial types, such as, S-LE370.

TRC-CONVR-TO-DATE-S

Used if the TO-DATE-MASK is one of the serial types, such as, S-LE370.

TRC-CONVR-OUT1-DATE-S

Used if the OUT1-DATE-MASK is one of the serial types, such as S-LE370.

TRC-CONVR-OUT2-DATE-S

Used if the OUT2-DATE-MASK is one of the serial types, such as S-LE370.

TRC-CONVR-OUT3-DATE-S

Used if the OUT3-DATE-MASK is one of the serial types, such as S-LE370.

TRC-CONVR-DB2-TABLE-ID (External Holiday Tables)

TRC-CONVR-DB2-TABLE-ID is a general use file name area set aside for customer use of DB2 tables or CICS VSAM files (by CICS file ID and/or data set name). The area is provided to the user with the CICS file ID set to TRCHLF01 (holiday CICS VSAM file is 01). The remainder of the area is set to spaces. No processing of this information is performed by CA Calendar Routines. Sample programs are provided as examples of using this area with site-specific modifications to CA Calendar Routines.

Reserved-for-User

The Reserved-for-User area has been set aside as an area where you can store data of any definition that can be included (or passed) with the CA Conversational Area. CA Calendar Routines will never use or modify this area.

CA-Only

Important! Do not modify the contents of this area.

An area has been set aside for CA processing. Three fields can be referenced to obtain the CA customer identification (CUSTOMER-ID), the product version number (VERSION-ID), and a field used by CA personnel to perform internal self-diagnosis (SELF-CHECK).

Chapter 3: Function Codes

This chapter provides a brief description of the types of function codes used by CA Calendar Routines.

This chapter includes the following sections:

- UTILITY functions- A brief description of the functions that perform specific tasks.
- BASIC functions- A brief description of the most common functions that you are likely to perform.
- EXTENDED function- A brief description of the functions that you are likely to use fairly often.
- CALENDAR-360 functions- A brief description of the functions that are specially designed for a Calendar that contains 360 Days.
- SPECIAL FISCAL CALENDAR functions- A brief description of the functions and related parameters that are specially designed for a Calendar that contains periods (similar to a month) of 4 or 5 weeks in duration.
- ADVANCED functions- A brief description of the functions that perform special or complex tasks.

UTILITY Functions

There are 13 UTILITY functions. These functions perform specific tasks.

- Each UTILITY function is specified by a one-character Function Type, followed by a three-digit Function Number.
- The Function Type for all UTILITY functions is the letter U.
- The Function Number for each UTILITY function will vary from 001 through 013, depending on the nature of your request.
  - U001 Current Date-Time
  - U002 Valid Date Y/N
  - U003 Leap Year Y/N
  - U004 Holiday Y/N
  - U005 Processing Day Y/N
  - U006 Identify Day of Week
  - U007 Date to Absolute Days
  - U008 Absolute Days to Date

- U009 Reformat Date
- U010 Age Function
- U011 Comparison Function
- U012 Century Concatenation
- U013 Century Truncation

Each of these UTILITY functions is described in the following sections.

(U001) CURRENT DATE-TIME

The CURRENT DATE-TIME function will provide the current date and time.

CURRENT DATE-TIME Parameters	
Business	1. OUT1-DATE-MASK
Input	1. FUNCTION-CODE
Output	1. OUT1-DATE 2. OUT-NUMERIC-PARM

Example 1	
Before Call	**After Call**
CCYYMMDD	
U001	
	20070517
	10562005

EXAMPLE 2	
Before Call	**After Call**
CCYYDDMM	
U001	
	20071705
	10562005

Note: The current date is May 17, 2007 and the time is 10562005.

- Example 1. The OUT1-DATE-MASK specifies CCYYMMDD. The FUNCTION-CODE requests current date and time, which CA Calendar Routines places in OUT1-DATE (in the format Century, Year, Month, Day) and OUT1-NUMERIC-PARM (in the format Hours, Minutes, Seconds and Hundredths of a second).
- Example 2. Same as above, except that, in example 2, the OUT1-DATE-MASK specifies CCYYDDMM, rather than CCYYMMDD.

(U002) VALIDATE DATE

The validate date function will decide if the specified date is valid. If the date is valid, then the answer is the value 1, otherwise the answer is the value 0.

VALIDATE DATE Parameters	
Business	1. From-Date-MASK
Input	1. FUNCTION-CODE
	2. From-Date
Output	1. RETURN-NUM
	2. OUT-NUMERIC-PARM

Example 1

Before Call	After Call
CCYYMMDD	
U002	
20070217	
	000
	1

Example 2

Before Call	After Call
CCYYMMDD	
U002	
200702W9	
	014
	0

- Example 1. The From-Date-MASK specifies CCYYMMDD. The FUNCTION-CODE requests validation. The From-Date, provided in CCYYMMDD format, is valid. Therefore, OUT-NUMERIC-PARM is 1.
- Example 2. Same as above, except that in example 2, the From-Date is not valid, because it is not numeric. Therefore, OUT-NUMERIC-PARM is 0. Also, notice that CA Calendar Routines has set the value of RETURN-NUM to 014, which designates “Bad From-Date.”

(U003) LEAP YEAR

The leap year function will decide if the specified date lies within a leap year. If it does, then the answer is the value 1, otherwise the answer is the value 0.

LEAP YEAR Parameters	
Business	1. From-Date-MASK
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	**After Call**
CCYYMMDD	
U003 20081225	
	1

Example 2

Before Call	**After Call**
MMCCYYDD	
U003 12200725	
	0

- Example 1. The From-Date-MASK specifies CCYYMMDD. The FUNCTION-CODE requests leap year status. The From-Date lies within the year 2008, which is a leap year. Therefore CA Calendar Routines sets OUT-NUMERIC-PARM to the value 1.
- Example 2. Same as above, except that in example 2, the From-Date is in MMCCYYDD format, and the year 2007 is not a leap year. Therefore, CA Calendar Routines sets OUT-NUMERIC-PARM to the value 0.

(U004) HOLIDAY

The holiday function will decide if the specified date is a holiday. If it is, then the answer is the value 1, otherwise the answer is the value 0.

HOLIDAY Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT-NUMERIC-PARM

Example 1	
Before Call	**After Call**
04 CCYYMMDD	
U004 20070704	
	1

Example 2	
Before Call	**After Call**
04 CCYYMMDD	
U004 20070203	
	0

Note: July 4, 2007 is in Holiday Table 04. February 3, 2007 is not in Holiday Table 04.

- Example 1. The HOLIDAY-TBL specifies Holiday Table 04. The From-Date-MASK specifies CCYYMMDD. The FUNCTION-CODE requests holiday status. The From-Date, July 4, 2007 is in Holiday Table 04. Therefore OUT-NUMERIC-PARM is 1.
- Example 2. Same as above, except that in example 2, the From-Date, February 3, 2007 is not in Holiday Table 04. Therefore OUT-NUMERIC-PARM is 0.

(U005) PROCESSING DAY

The processing day function will decide if the specified date is a processing day. If it is, then the answer is the value 1, otherwise the answer is the value 0.

PROCESSING DAY Parameters	
Business	1. HOLIDAY-TBL 2. PROC-DAY-DEF 3. From-Date-MASK
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	After Call
05 'AAAAAAA' CCYYMMDD	
U005 20070216	
	1

Example 2	
Before Call	**After Call**
05 'NEEEEEN' CCYYMMDD	
U005 20070216	
	0

Note: Data Stipulations: February 16, 2007 is in Holiday Table 05 and is a Friday.

- Example 1. The HOLIDAY-TBL specifies table number 05. The From-Date-MASK specifies CCYYMMDD. The PROC-DAY-DEF defines the processing day for each day of the week. The FUNCTION-CODE requests processing day status. The From-Date, February 16, 2005 is in Holiday Table 05; and, furthermore, it falls on a Friday. According to PROC-DAY-DEF, Friday is a processing day, always. Therefore, CA Calendar Routines considers the From-Date a processing day; so it sets OUT-NUMERIC-PARM to the value 1.
- Example 2. Same as above, except that in example 2, Friday is specified as a processing day, excluding holidays. Therefore CA Calendar Routines does not consider the From-Date a processing day; so it sets OUT-NUMERIC-PARM to the value 0.

(U006) DAY OF WEEK

The day of week function will identify the day of the week that corresponds to the specified date. The answer ranges from the numeral specified for Sunday in the DOW-STRING to the numeral specified for Saturday in the DOW-STRING.

DAY OF WEEK Parameters	
Business	1. From-Date-MASK 2. DOW-STRING
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT-NUMERIC-PARM 2. OUT1,2,3-DATE-E

Example 1	
Before Call	**After Call**
CCYYMMDD 1234567	
U006 20070219	
	2 Monday

Example 2	
Before Call	**After Call**
CCYYMMDD 0123456	
U006 20070516	
	3 Wednesday

Note: Data Stipulations: February 19 2007 falls on a Monday; May 16, 2007 falls on a Wednesday.

- Example 1. The From-Date-MASK specifies CCYYMMDD. The DOW-STRING specifies the numerals 1 through 7. The FUNCTION-CODE requests day of the week. The From-Date, February 19, 2007 falls on a Monday. Therefore, CA Calendar Routines sets OUT-NUMERIC-PARM to the value 2, and the literal Monday will appear in the output-dates.
- Example 2. Same as above, except that, in example 2, the DOW-STRING specifies the numerals 0 through 6, and May 16, 2007 is a Wednesday. Therefore, CA Calendar Routines sets OUT-NUMERIC-PARM to the value 3, and the literal Wednesday appears in the output-dates.

(U007) DATE --> ABSOLUTE DAYS

The absolute days function will convert the specified date into the absolute day count. The absolute day count is the number of days since January 01, 0001.

DATE-->ABSOLUTE DAYS Parameters	
Business	1. From-Date-MASK
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	After Call
CCYYMMDD	
U007 00010101	
	1

Example 2

Before Call	After Call
CCYYMMDD	
U007 99991231	
	3652056

- Example 1. The From-Date-MASK specifies CCYYMMDD. The FUNCTION-CODE requests absolute day count. The From-Date, January 01, 0001 corresponds to day 1. Therefore, CA Calendar Routines sets OUT-NUMERIC-PARM to the value 1.
- Example 2. Same as above, except, that in example 2, the date 99991231 corresponds to day 3652056.

(U008) ABSOLUTE DAYS --> DATE

The absolute days function will convert the specified absolute day count into the corresponding calendar date.

ABSOLUTE DAYS--> DATE Parameters	
Business	1. OUT1-DATE-MASK
Input	1. FUNCTION-CODE 2. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example 1

Before Call	After Call
CCYYMMDD	
U008 1	
	00010101

Example 2

Before Call	After Call
CCYYMMDD	
U008 +3652056	
	99991231

- Example 1. The OUT1-DATE-MASK specifies CCYYMMDD. The FUNCTION-CODE requests Calendar date. The IN-NUMERIC-PARM, specifies day count 1. Therefore, CA Calendar Routines sets OUT1-DATE to January 01, 0001.
- Example 2. Same as above, except that in example 2, the IN-NUMERIC-PARM specifies day count 3652056. Therefore, CA Calendar Routines sets OUT1-DATE to 99991231.

(U009) REFORMAT DATE

The reformat date function will convert the specified From-Date into the specified STDOUT, OUT1, OUT2, and OUT3 dates.

Note: With this Utility function you can convert among all the date formats; Gregorian, Julian, Numeric, Alphanumeric, Edited, Calendar-360, 9's complement or Serial.

REFORMAT DATE Parameters	
Business	1. From-Date-MASK
	2. OUT1-DATE-MASK
	3. OUT2-DATE-MASK
	4. OUT3-DATE-MASK
Input	1. FUNCTION-CODE
	2. From-Date
Output	1. STDOUT-DATE
	2. OUT1-DATE
	3. OUT2-DATE
	4. OUT3-DATE

Example 1

Before Call	After Call
CCYYMMDD	
CCYYDDMM	
YYMMDD–	
-CCYYDDD	
U009	
20080301	
	20080301
	20080103
	080301
	02008061

Example 2	
Before Call	**After Call**
CCYYMMDD CCYYDDMM YYMMDD– -CCYYDDD	
U009 20070301	
	20070301 20070103 070301 02007060

- Example 1. CA Calendar Routines converts the From-Date, in a format specified by the From-Date-MASK to the various output dates, whose formats are specified by their respective MASKS. Since 2008 is a leap year, March 1 corresponds to Julian Day 61.
- Example 2. Same as above, except that, in example 2, the year 2007 is not a leap year. Therefore March 1 corresponds to Julian Day 60.

(U010) AGE FUNCTION

The age function will allow you to calculate the age in years, months, and days between two dates. The From-Date is the earlier date (presumably the birth date), and the To-Date is the later date (presumably the date as of when the calculation should be made). The To-Date should not be less than the From-Date, otherwise you will get error code 42. The ENFORCE or NO-ENF parameter will count months and years either enforcing the end of month boundaries or not.

AGE FUNCTION Parameters	
Business	1. From-Date-MASK 2. To-Date-MASK
Input	1. FUNCTION-CODE 2. From-Date 3. To-Date 4. IN-ALPHA-PARM
Output	1. OUT-NUMERIC-PARM

Example 1	
Before Call	**After Call**
CCYYMMDD	
CCYYMMDD	
U010	
19791231	
20060912	
ENFORCE	
	+00260812

Example 2	
Before Call	**After Call**
CCYYMMDD	
CCYYMMDD	
U010	
19791231	
20070912	
NO-ENF	
	+00270815

- Example 1. CA Calendar Routines starts on Dec 31, 1979 and jumps ahead to Dec 31, 2005. (26 years). Then it travels by month-end through Jan 2006, Feb 2006, and so on, until Aug 31, 2006. (eight months). Then it travels 12 days, hence the answer +00260812. (0026 years 08 months and 12 days).
- Example 2. Same as above, except that, when it gets to Feb 28, 2007 it goes to Mar 28th and then Apr 28 (for example does not follow the month-end enforcement), so that when it gets to Aug 28, it is 15 days from Sep 12 rather than the previous example of being 12 days away from Sep 12. Thus the final answer is +00270815. (0027 years 08 months and 15 days).

Note: The value in OUT-NUMERIC-PARM is not a date. It contains years, months, and days, but any of those values could be zero if the duration is zero—and zero is not a valid year, month, or day.

(U011) COMPARISON FUNCTION

The comparison function allows you to compare two dates to each other. The function also provides additional comparison of data which is not a valid date (the From-Date / To-Date must have matching formats for non-date comparisons):

```
YY------, YYMM----, ------YY, ----YYMM, CCYY----, ----CCYY
```

This function will use the CENTURY-BREAK-FEATURE to concatenate the century to the beginning of the provided dates, so that they can be compared. Editing of the dates themselves is *not* performed and output dates are *not* provided. This function was created to support the windowing option for field expansion.

Note: COBOL language 88s are now provided to test for GREATER-THAN, LESS-THAN, or EQUAL-TO.

COMPARISON FUNCTION Parameters	
Business	1. From-Date-MASK
	2. To-Date-MASK
Input	1. FUNCTION-CODE
	2. CENTURY-BREAK
	3. From-Date
	4. To-Date
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	**After Call**
YYMMDD— CCYYMMDD	
U011 19/20;68 991215 20000115	
	-00000001

Example 2	
Before Call	**After Call**
---YYDDD	
--YYMMDD	
U011	
SLIDE-20	
01005	
970301	
	+00000001

- Example 1. In this alpha-Gregorian example, 991215 is converted to 19991215 and compared to 20000115. The answer -00000001 means that 19991215 is less than 20000115.
- Example 2. In this numeric-Julian example, 01005 is converted to 2001005 and 970301 is converted to 19970301. The answer +00000001 means that Julian date 2001005 (1-1-2001) is greater than19970301 (3-1-1997). The current computer date for this example was 12-10-1997 setting the “sliding window” century:
  - YEAR VALUE 77 through 99 are assigned CENTURY 19
  - YEAR VALUE 00 through 76 are assigned CENTURY 20

(U012) CONCATENATE THE CENTURY FUNCTION

The concatenate the century function concatenates the century to the beginning or end of the provided date formats. Editing of the input (from) date is *not* performed. If you need date validation processing with century insertion, use function U009. The output date is stored in OUT1, OUT2, and OUT3. Julian, Gregorian, and the 360-day Calendar date formats are supported by this function.

From-Date-Mask	Yields OUT 1,2,3 DATEX
--YYMMDD	CCYYMMDD
YYMMDD--	CCYYMMDD
YYDDD---	CCYYDDD-
---YYDDD	-CCYYDDD
---YYKKK	-CCYYKKK
YYKKK---	CCYYKKK-
----YYMM	--CCYYMM

From-Date-Mask	Yields OUT 1,2,3 DATEX
YYMM----	CCYYMM--
------YY	----CCYY
YY------	CCYY----

CONCATENATE THE CENTURY FUNCTION Parameters	
Business	1. From-Date-MASK
Input	1. FUNCTION-CODE
	2. CENTURY-BREAK
	3. From-Date
Output	1. OUT1-DATE
	2. OUT2-DATE
	3. OUT3-DATE

Example 1

Before Call	After Call
YYMMDD-	
U012	
19/20;68	
070903	
	20070903
	20070903
	20070903

Example 2

Before Call	After Call
---YYDDD	
U012	
SLIDE-20	
071	
	200781
	2007181
	2007181

- Example 1. In this alpha-Gregorian example, the 20th century is placed in front of date 070903 and stored in OUT1, OUT2, and OUT3 alphanumeric date areas.
- Example 2. In this alpha-Julian example, the 20th century is placed in front of date 07181 and stored in OUT1, OUT2, and OUT3 alphanumeric date areas. The current computer date for this example was 12-10-2007 setting the sliding window century:
  - YEAR VALUE 77 through 99 are assigned CENTURY19
  - YEAR VALUE 00 through 76 are assigned CENTURY 20

(U013) TRUNCATE THE CENTURY FUNCTION

The truncate the century function truncates the two-character century from the beginning or end of the provided date formats. Editing of the input (from) date is not performed. If you need date validation processing with century truncation, use function U009. The output date is stored in OUT1, OUT2, and OUT3. Julian, Gregorian, and 360-day Calendar date formats are supported by this function.

From-Date-Mask	Out 1-Date-Mask*	Out 1,2,3	Date Type
CCYYMMDD	--YYMMDD	00YYMMDD	(numeric)
CCYYMMDD	YYMMDD-	YYMMDD	(alpha)
-CCYYDDD		000YYDDD	(numeric)
-CCYYKKK		000YYKKK	(numeric)
CCYYDDD		YYDDD	(alpha)
CCYYKKK		YYKKK	(alpha)
--CCYYMM		0000YYMM	(numeric)
----CCYY		000000YY	(numeric)
CCYYMM-		YYMM	(alpha)
CCYY---		YY	(alpha)

Note: For From-Date-MASK “CCYYMMDD”, the OUT1-DATE-MASK must contain “YYMMDD--” (alpha-numeric format output) or “--YYMMDD” (numeric format output).

TRUNCATE THE CENTURY FUNCTION Parameters	
Business	1. From-Date-MASK
	1. OUT1-DATE-MASK
Input	1. FUNCTION-CODE
	2. CENTURY-BREAK
	3. From-Date
Output	1. OUT1-DATE
	2. OUT2-DATE
	3. OUT3-DATE

Example 1

Before Call	After Call
CCYYMMDD	
YYMMDD-	
U013	
19/20;68	
20070903	
	070903
	070903
	070903

Example 2

Before Call	After Call
CCYYDDD	
U013	
SLIDE-20	
2007181	
	07181
	07181
	07181

- Example 1. In this alpha-Gregorian example, the 20th century is removed from the front of date 20070903 and stored in OUT1, OUT2, and OUT3 alphanumeric date areas.
- Example 2. In this alpha-Julian example, the 20th century is removed from the front of date 2007181 and stored in OUT1, OUT2, and OUT3 alphanumeric date areas. The current computer date for this example was 12-10-2007 setting the sliding window" century:
  - YEAR VALUE 77 through 99 are assigned CENTURY19
  - YEAR VALUE 00 through 76 are assigned CENTURY 20

BASIC Functions

The BASIC functions perform the most rudimentary calculations; "Days Between" and "Date +/- N Days, Weeks, Months and Years."

Each BASIC function is specified by a one-character Function Type, followed by a three-digit Function Number.

- The Function Type for the BASIC Functions is either C or P.
  - C means that CA Calendar Routines considers all calendar days.
  - P means that CA Calendar Routines considers only processing days.
- The Function Number for BASIC functions, is either 101, 102, or 103. This depends on the type of calculation needed by the program.
  - 101 Days Between
  - 102 Date +/- N days
  - 103 Date +/- N weeks, months or years

Each of these BASIC functions is described in their respective sections.

(C/P 101) DAYS BETWEEN

The days between function will count the number of days from the From-Date to the To-Date. If the From-Date is earlier than the To-Date, the answer is positive. If the From-Date is later than the To-Date, the answer is negative. If the two dates are equal, then the answer is 0.

Examples 1 and 2

DAYS BETWEEN Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. To-Date-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. To-Date
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	After Call
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C101	
20070903	
200707	
	+4

Example 2	
Before Call	**After Call**
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P101	
20070903	
20070907	
	+1

Note: Data Stipulation: September 6, 2007 is in Holiday Table 06 and is a Monday.

Our goal is to count the number of days from the From-Date to the To-Date. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In Example 1, every day qualifies; and in Example 2, only processing days qualify.

Example	Date	Qualification	Count
Example 1	20070903	N/A	+0
	20070904	YES	+1
	20070905	YES	+2
	20070906	YES	+3
	20070907	YES	+4
Example 2	20070903	N/A	+0
	20070904	NO, Saturdays are never processed.	+0
	20070905	NO, Sundays are never processed.	+0
	20070906	NO, Monday-Holidays are not processed.	+0
	20070907	YES	+1

Examples 3 and 4

DAYS BETWEEN Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. To-Date-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. To-Date
Output	1. OUT-NUMERIC-PARM

Example 3

Before Call	After Call
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C101	
20070907	
20070903	
	-4

Example 4

Before Call	After Call
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P101	
20070907	
20070903	
	-1

Note: Data Stipulation: September 6, 2007 is in Holiday Table 06 and is a Monday.

Examples 3 and 4 are similar to Examples 1 and 2, except that the From-Date and the To-Date have been reversed

Example	Date	Qualification	Count
Example 3	20070907	N/A	-0
	20070906	YES	-1
	20070905	YES	-2
	20070904	YES	-3
	20070903	YES	-4
Example 4	20070907	N/A	-0
	20070906	NO, Monday-Holidays are not processed.	-0
	20070905	NO, Sundays are never processed.	-0
	20070904	NO, Saturdays are never processed.	-0
	20070903	YES	-1

(C/P 102) DATE +/- *n* DAYS

The date +/- *n* days function will calculate the date that is *n* days later than or earlier than the specified date.

Examples 1 and 2

DATE +/- *n* DAYS Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. OUT1-DATE-MASK 4. PROC-DAY-DEF 5. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example 1

Before Call	After Call
06 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C102 20070903 +4	
	20100907

Example 2

Before Call	After Call
06 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
P102 20100903 +4	
	20100910

Note: Data Stipulation: September 6, 2010 is in Holiday Table 06 and it is a Monday.

Our goal is to calculate the date that is +4 days from the From-Date. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In example 1, every day qualifies; and in example 2, only processing days qualify

Example	Date	Qualification	count
Example 1	20100903	N/A	+0
	20100904	YES	+1
	20100905	YES	+2
	20100906	YES	+3
	20100907	YES	+4
Example 2	20100903	N/A	+0
	20100904	NO, Saturdays are never processed.	+0
	20100905	NO, Sundays are never processed.	+0

Example	Date	Qualification	count
	20100906	NO, Monday-Holidays are not processed.	+0
	20100907	YES	+1
	20100908	YES	+2
	20100909	YES	+3
	20100910	YES	+4

DATE +/- *n* DAYS Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. OUT1-DATE-MASK 4. PROC-DAY-DEF 5. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example 3

Before Call	After Call
06 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C102 20100907 -4	
	20100903

Example 4	
Before Call	**After Call**
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P102	
20100910	
-4	
	20100903

Note: Data Stipulation: September 6, 2010 is in Holiday Table 06 and it is a Monday.

Examples 3 and 4 are similar to examples 1 and 2, except that the direction is toward the past, rather than toward the future. (–4 rather than +4)

Example	Date	Qualification	count
Example 3	20100907	N/A	-0
	20100906	YES	-1
	20100905	YES	-2
	20100904	YES	-3
	20100903	YES	-4
Example 4	20100910	N/A	-0
	20100909	YES	-1
	20100908	YES	-2
	20100907	YES	-3
	20100906	NO, Monday-Holidays are not processed.	-3
	20100905	NO, Sundays are never processed.	-3
	20100904	NO, Saturdays are never processed.	-3
	20100903	YES	-4

(C/P 103) DATE +/-*n* WEEKS, MONTHS or YEARS

This function will calculate the date that is *n* weeks, months, or years later than or earlier than the specified date.

Weeks: Examples 1 and 2

DATE +/- *n* WEEKS, MONTHS or YEARS Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
	6. IN-ALPHA-PARM
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example 1

Before Call	**After Call**
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
'WEEKS'	
C103	
20100308	
+2	
	20100322

Example 2

Before Call	After Call
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
'WEEKS'	
P103	
20100927	
-3	
	20100903

Note: Data Stipulation: September 6, 2010 is in Holiday Table 06 and it is a Monday.

- Example 1. The goal is to calculate the date that is +2 weeks from the From-Date. The first step is to convert 2 weeks into 14 days. The date that is 14 days forward from 20100308 is 20100322.
- Example 2. The goal is to calculate the date that is -3 weeks from the From-Date. The first step is to convert 3 weeks into 21 days. The date that is 21 days backward from 20100927 is 20100906. However, since 20100906 is a holiday, CA Calendar Routines continues until a processing day is reached. That day turns out to be 20100903, since Saturday and Sunday are not processed.

Months and Years: Examples 3, 4, 5, 6

When durations of months and years are calculated, CA Calendar Routines uses the concept of end of month enforcement. Consider the example of May 31 plus one month. The answer is June 30. Then, if one month is subtracted from June 30, the answer would be May 30. Thus May 31 +1 month -1 month does not equal May 31. That is, unless the end of month rule is enforced. Using this rule it is possible to subtract one month from June 30 and yield May 31. When months and years are the duration in CA Calendar Routines, you must use the IN-ALPHA-PARM of M-ENF (Month-Enforce), M-NOENF (Month-No-Enforce), Y-ENF (Year-Enforce), or Y-NOENF (Year-No-Enforce). For yearly calculations the principle is similar. February 28, 2010 minus one year (enforce) is February 29 2009; whereas February 28, 2008 minus one year (no-enforce) is February28 2007.

DATE + / - *n* WEEKS, MONTHS or YEARS Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
	6. IN-ALPHA-PARM
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example 1

Before Call	After Call
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
'M-NOENF'	
C103	
20100630	
-3	
	20100330

Example 2	
Before Call	**After Call**
06	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
'M-NOENF'	
P103	
20100806 +1	
	20100907

Note: Data Stipulation: September 6, 2010 is in Holiday Table 06 and it is a Monday.

- Comments Example 3. We start on June 30 which is the end of the month but we do not want to enforce it. Therefore when we get to March 30 as the nominal answer we stay there rather than forcing it to March 31.
- Comments Example 4. We start on August 6. We have specified no-enforce, but it is a principle that would only apply if the start date was the end of the month. When we get to September 6, however, we discover that it is a holiday and work forwards until we reach a processing day which is September 7.

DATE + / - N WEEKS, MONTHS or YEARS Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
	4. IN-ALPHA-PARM
Output	1. OUT1-DATE

Example 1

Before Call	After Call
05 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C103 20100228 +1 'Y-ENF'	
	20110229

Example 2

Before Call	After Call
05 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
P103 20141231 -4 'Y-NOENF'	
	20101230

Note: Data Stipulation: December 31, 2010 is in Holiday Table 05 and it is a Friday.

- Comments Example 5. We start on February 28 2010. Since 2010 is not a leap year, February 28 is the end of the month and we want to enforce it. Therefore when we get to February 28, 2011 we must adjust the date to the 29 which is the last day of February in 2011.
- Comments Example 6. We start on December 31 2014, and work our way back to December 31 2010. In this case there is no need to enforce the end of month rule since the new month's last day is the same as the starting month's last day. However, December 31, 2010 is a holiday, so we work backwards until we reach a processing day which is December 30, 2010.

EXTENDED Functions

There are fourteen EXTENDED functions. These functions are extensions of the BASIC functions. They are designed to answer common questions, such as: “What is the next day?” or “What date is three days before the end of the month?”

Each EXTENDED function is specified by a one-character Function Type, followed by a three-digit Function Number.

- The Function Type for all EXTENDED functions is either C or P.
  - C means that CA Calendar Routines considers all calendar days.
  - P means that CA Calendar Routines considers only processing days.
- The Function Number for EXTENDED functions ranges from 201 through 207. The choice depends on the type of calculation needed by the application program.
  - 201 Next Day
  - 202 Previous Day
  - 203 Days Since Beginning of Month
  - 204 Date Until End of Month
  - 205 Beginning of Month +/- *n* days
  - 206 End of Month +/- *n* days
  - 207 *n*th DOW Occurrence Forward/Backward

Each of these EXTENDED functions is described in the following sections.

(C/P 201) NEXT DAY

This function will provide the date that is one day later than the date that you provide.

NEXT DAY Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
Output	1. OUT1-DATE

Example 1

Before Call	After Call
07	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C201	
20100212	
	20100213

Example 2

Before Call	After Call
07	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P201	
20100212	
	20100216

Note: Data Stipulation: February 15, 2010 is in Holiday Table 07 and it is a Monday.

Our goal is to find the calendar date that is one day later than February 12, 2010. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In Example 1, every day qualifies; and in Example 2, only processing days qualify.

Example	Date	Qualification	count
Example 1	20100212	N/A	+0
	20100213	YES	+1
Example 2	20100212	N/A	+0
	20100213	NO, Saturdays are never processed.	+0
	20100214	NO, Sundays are never processed.	+0
	20100215	NO, Monday-Holidays are not processed.	+0
	20100216	YES	+1

(C/P 202) PREVIOUS DAY

This function will find the date that is one day earlier than the date that you provide.

PREVIOUS DAY Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
Output	1. OUT1-DATE

Example 1

Before Call	After Call
07 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C202 20100216	
	20100215

Example 2

Before Call	After Call
Before Call	After Call
07 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
P202 20100216	20100212

Note: Data Stipulation: February 15, 2007 is in Holiday Table 07 and it is a Monday.

Our goal is to find the calendar date that is one day earlier than February 16, 2010. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In example 1, every day qualifies; and in example 2, only processing days qualify.

Example	Date	Qualification	Count
Example 1	20100216	N/A	-0
	20100215	YES	-1
Example 2	20100216	N/A	-0
	20100215	NO, Monday-Holidays are not processed.	-0
	20100214	NO, Sundays are never processed.	-0
	20100213	NO, Saturdays are never processed.	-0
	20100212	YES	-1

(C/P 203) DAYS SINCE THE BEGINNING OF THE MONTH

This function will provide the number of days from the date that you specify to the beginning of the month.

DAYS SINCE THE BEGINNING OF THE MONTH Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. PROC-DAY-DEF 4. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	After Call
05 CCYYMMDD 'NEEEEEN' 'T'	
C203 20110706	
	-5

Example 2

Before Call	After Call
05 CCYYMMDD 'NEEEEEN' 'T'	
P203 20110706	

Example 2	
Before Call	**After Call**
	-2

Note: Data Stipulation: July 04, 2011 is in Holiday Table 05 and it is a Monday.

Our goal is to calculate the number of days from July 06, 2011 back to the beginning of the month. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In example 1, every day qualifies; and in example 2, only processing days qualify.

Example	Date	Qualification	Count
Example 1	20110706	N/A	-0
	20110705	YES	-1
	20110704	YES	-2
	20110703	YES	-3
	20110702	YES	-4
	20110701	YES	-5
Example 2	20110706	N/A	-0
	20110705	YES	-1
	20110704	NO, Monday-Holidays are not processed.	-1
	20110703	NO, Sundays are never processed.	-1
	20110702	NO, Saturdays are never processed.	-1
	20110701	YES	-2

(C/P 204) DAYS UNTIL THE END OF THE MONTH

This function will provide the number of days from the date that you specify to the end of the month.

DAYS UNTIL THE END OF THE MONTH Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. PROC-DAY-DEF 4. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date

DAYS UNTIL THE END OF THE MONTH Parameters	
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	After Call
08 CCYYMMDD 'NEEEEEN' 'T'	
C204 20091223	
	+8

Example 2

Before Call	After Call
08 CCYYMMDD 'NEEEEEN' 'T'	
P204 20091223	
	+5

Note: Data Stipulation: December 25, 2009 is in Holiday Table 08 and it is a Friday.

Our goal is to calculate the number of days from December 23, 2009 forward to the end of the month. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In example 1, every day qualifies; and in example 2, only processing days qualify.

Example	Date	Qualification	Count
Example 1	20091223	N/A	+0
	20091224	YES	+1
	20091225	YES	+2
	20091226	YES	+3
	20091227	YES	+4
	20091228	YES	+5
	20091229	YES	+6
	20091230	YES	+7

Example	Date	Qualification	Count
	20091231	YES	+8
Example 2	20091223	N/A	+0
	20091224	YES	+1
	20091225	NO, Friday-Holidays are not processed.	+1
	20091226	NO, Saturdays are never processed.	+1
	20091227	NO, Sundays are never processed.	+1
	20091228	YES	+2
	20091229	YES	+3
	20091230	YES	+4
	20091231	YES	+5

(C/P 205) BEGINNING OF THE MONTH +/- N DAYS

This function will provide the date that is *n* days before or after the beginning of the month specified by the input date.

BEGINNING OF THE MONTH +/- N DAYS Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. OUT1-DATE-MASK 4. PROC-DAY-DEF 5. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example 1	
Before Call	**After Call**
05 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C205 20110723 +5	
	20110706

Example 2	
Before Call	**After Call**
05 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
P205 20110723 +5	
	20110711

Note: Data Stipulation: July 04, 2011 is in Holiday Table 05 and it is a Monday.

We want to find the date that is five days after the beginning of the month containing July 23, 2011. First, we define a new From-Date as the beginning of July; (July 01, 2011). Then, according to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In example 1, every day qualifies; and in example 2, only processing days qualify.

Example	Date	Qualification	Count
Example 1	20110701	N/A	+0
	20110702	YES	+1
	20110703	YES	+2
	20110704	YES	+3
	20110705	YES	+4
	20110706	YES	+5

Example	Date	Qualification	Count
Example 2	20110701	YES	+6
	20110702	YES	+7
	20110703	YES	+8
	20110704	N/A	+0
	20110705	YES	+1
	20110706	NO, Friday-Holidays are not processed.	+1
	20110707	NO, Saturdays are never processed.	+1
	20110708	NO, Sundays are never processed.	+1
	20110709	YES	+2
	20110710	YES	+3
	20110711	YES	+4

(C/P 206) END OF THE MONTH +/- *n* DAYS

This function will provide the date that is *n* days before or after the end of the month specified by the input date.

END OF THE MONTH +/- *n* DAYS Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. OUT1-DATE-MASK 4. PROC-DAY-DEF 5. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example 1

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C206 20091205 -4	
	20091227

Example 2

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
P206 20091218 -4	
	20091224

Note: Data Stipulation: December 25, 2009 is in Holiday Table 08 and it is a Friday.

We want to find the calendar date that is four days prior to the end of the month containing December 18, 2009. First, we define a new From-Date as the end of the month (December 31, 2009). Then, according to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In example 1, every day qualifies; and in example 2, only processing days qualify.

Example	Date	Qualification	Count
Example 1	20091231	N/A	-0
	20091230	YES	-1
	20091229	YES	-2
	20091228	YES	-3
	20091227	YES	-4
Example 2	20091231	N/A	-0
	20091230	YES	-1
	20091229	YES	-2
	20091228	YES	-3
	20091227	NO, Sundays are never processed.	-3
	200912	NO, Saturdays are never processed.	-3

Example	Date	Qualification	Count
	20091231	N/A	-0
	26		
	20091225	NO, Friday holidays are not processed.	-3
	20091224	YES	-4

(C/P 207) Nth DOW Occurrence Forward/Backward

This function provides the date that is the *n*th day forward or backward from the reference date (while taking into account the specified days of the week).

Nth DOW Occurrence Forward/Backward Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. OUT1-DATE-MASK 4. PROC-DAY-DEF
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM 4. IN-ALPHA-PARM
Output	1. OUT1-DATE

Example 1

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN'	
C207 20091228 -2 'NNNNNYN'	
	20091218

Example 2

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN'	

Example 2	
Before Call	**After Call**
P207 20091228 -2 'NNNNNYN'	
	20091211

Note: Data Stipulation: December 25, 2009 is in Holiday Table 08 and it is a Friday.

We want to find the calendar date that is the second Friday prior to the date that is specified in the From-Date, December 28, 2009. In example 1, every Friday qualifies; and in example 2, only processing Fridays qualify.

Example	Date	Qualification	Count
Example 1	20091228	N/A	-0
	20091225	YES	-1
	20091218	YES	-2
Example 2	20091228	N/A	-0
	20091225	NO, Friday holidays are not processed.	-0
	20091218	YES	-1
	20091211	YES	-2

CALENDAR-360 Functions

There are four CALENDAR-360 Functions. These functions perform calculations on dates that lie within a year that contains 360 fictitious days (as opposed to a year that contains the usual 365 or 366 days). The goal of working with a calendar that contains 360 days is to create 12 equal months of 30 days each. Months containing 31 days such as January and March will be shortened to contain 30 days. The 30 and 31 of those months will have the same 360-day designation. For example, December 30 and December 31 are both converted to day 360 on the 360-Day Calendar. Likewise, adjustments are made for February to normalize it to a 30-day month (for leap year and non-leap year).

- The Function Type for every CALENDAR-360 function is K.
  - K means CALENDAR-360 Year.
- The Function Number for CALENDAR-360 functions will vary from 301 to 304 depending on the nature of your request.
  - K301 Convert a Calendar-365/366 Date to Calendar-360 Format
  - K302 Convert a Calendar-360 Date to Calendar-365/366 Format
  - K303 Calculate the number of days between two Calendar-360 Dates
  - K304 Calculate a Calendar-360 Date that is +/- *n* Days, Months or Years from an input Calendar-360 Date.

Dates on the 360-calendar are specified using a special set of Date-Masks where CC means century, YY means year, and KKK means day (001 through 360) Date-masks are described under From-Date-MASK (To-Date-MASK, OUT1-DATE-MASK, OUT2-DATE-MASK and OUT3-DATE-MASK) .

Each of the four CALENDAR-360 functions is described in detail in the following sections.

(K301) CONVERT A 365/366 DATE TO 360 FORMAT

This function will enable you to convert a date on the 365/366-calendar to the corresponding date on the 360-calendar.

CONVERT A 365/366 DATE TO 360 FORMAT Parameters	
Business	1. From-Date-MASK 2. OUT1-DATE-MASK
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT1-DATE

Example 1

Before Call	After Call
MMDDCCYY CCYYKKK	
K301 08252011	
	2011235

Example 2	
Before Call	**After Call**
CCYYDDD-CCYYKKK-	
K301 2011237	
	2011235

- Example 1. The input date on the 365/366-Calendar year is specified in MMDDCCYY format. The function, K301, calls for conversion to a 360-day format. The output format is correctly specified as one of the special Date-Masks reserved for 360-Day Calendar Dates, CCYYKKK-.
- Example 2. The input date on the 365/366-calendar year is specified in CCYYDDD- format. The function, K301, calls for conversion to a 360-day format. The output format is correctly specified as one of the special Date-Masks reserved for 360-Day Calendar Dates, CCYYKKK-.

(K302) CONVERT A 360 DATE TO 365/366 FORMAT

This function will enable you to convert a date on the 360-calendar to the corresponding date on the 365/366-calendar.

CONVERT A 360 DATE TO 365/366 FORMAT Parameters	
Business	1. From-Date-MASK 2. OUT1-DATE-MASK
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT1-DATE

Example 1	
Before Call	**After Call**
CCYYKKK- MMDDCCYY	
K302 2011237	
	08272011

Example 2	
Before Call	**After Call**
CCYYKKK- CCYYDDD-	
K302 2011235	
	2011237

- Example 1 The input date on the 360-calendar year is correctly specified as one of the special Date-Masks reserved for 360-Day Calendar Dates (CCYYKKK-). Function K302 requests conversion to a 365/366-day format.
- Example 2 The input date on the 360-calendar year is correctly specified as one of the special Date-Masks reserved for 360-Day Calendar Dates (CCYYKKK-). Function K302 requests conversion to a 365/366-day format.

(K303) DAYS BETWEEN TWO 360-DAY CALENDAR DATES

This function will enable you to calculate the number of days between two dates on the 360-day calendar.

DAYS BETWEEN TWO 360-DAY CALENDAR DATES Parameters	
Business	1. From-Date-MASK 2. To-Date-MASK
Input	1. FUNCTION-CODE 2. From-Date 3. To-Date
Output	1. OUT-NUMERIC-PARM

Example 1	
Before Call	**After Call**
CCYYKKK- -CCYYKKK	
K303 2011200 02011235	
	+35

Example 2	
Before Call	**After Call**
---KKKYY KKKYY--	
K303 00020094 23594	
	+35

- Example 1 The From-Date and the To-Date are both correctly specified as special Date-Masks reserved for 360-Day Calendar Dates. Function K303 asks for the number of days from the From-Date to the To-Date.
- Example 2 This a repeat of Example 1, except that different Date-Masks have been specified.

DAYS BETWEEN TWO 360-DAY CALENDAR DATES Parameters	
Business	1. From-Date-MASK 2. To-Date-MASK
Input	1. FUNCTION-CODE 2. From-Date 3. To-Date
Output	1. OUT-NUMERIC-PARM

Example 3	
Before Call	**After Call**
CCYYKKK- CCYYKKK-	
K303 2011175 2010291	
	+476

Example 4	
Before Call	**After Call**
CCYYKKK-	
CCYYKKK-	
K303	
2010291	
2011175	
	-476

- Example 3. The From-Date and the To-Date are both correctly specified as special Date-Masks reserved for 360-Day Calendar Dates. Function K303 asks for the number of days from the From-Date to the To-Date.
- Example 4. This is a repeat of Example 3, except that the dates have been reversed.

(K304) 360-DAY DATE +/- *n* DAYS, MONTHS OR YEARS

This function will enable you to start with a date on the 360-day calendar and add (or subtract) a specified number of days, months or years. These durations are specified as K360-DY, K360-MO, or K360-YR.

360-DAY DATE +/- *n* DAYS, MONTHS OR YEARS Parameters	
Business	1. From-Date-MASK
	2. OUT1-DATE-MASK
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
	4. IN-ALPHA-PARM
Output	1. OUT1-DATE

Example 1	
Before Call	**After Call**
CCYYKKK-	
CCYYKKK-	

Example 1	
Before Call	**After Call**
K304 2011200 +351 K360-DY	
	2010191

Example 2	
Before Call	**After Call**
CCYYKKK- CCYYKKK-	
K304 2010191 -351 K360-DY	
	2011200

- Example 1. The From-Date and the OUT1-DATE are both correctly specified as special Date-Masks reserved for 360-Day Calendar Dates. Function K304 requests that 351 days be added to the From-Date. The answer is in the OUT1-DATE.
- Example 2. This is a repeat of Example 1, except that we have requested that 351 days be subtracted from the From-Date.

360-DAY DATE + / - *n* DAYS, MONTHS OR YEARS Parameters	
Business	1. From-Date-MASK 2. OUT1-DATE-MASK
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM 4. IN-ALPHA-PARM
Output	1. OUT1-DATE

Example 3

Before Call	After Call
CCYYKKK- CCYYKKK-	
K304 2011200 +2 K360-MO	
	2011260

Example 4

Before Call	After Call
CCYYKKK-CCYYKKK-	
K304 1995151 -3 K360-YR	
	2009151

- Example 3. The From-Date and the OUT1-DATE are both correctly specified as special Date-Masks reserved for 360-Day Calendar Dates. Function K304 requests that two months be added to the From-Date. The answer is in the OUT1-DATE (60 days from the From-Date).
- Example 4. This is similar to Example 3, except that we would like to subtract three years from From-Date. The answer is found in OUT1-DATE.

SPECIAL FISCAL CALENDAR Functions

Special Fiscal Calendar functions perform conversions and calculations on dates which are defined in special fiscal calendars. See Special Fiscal Calendars for the definition of the calendars and date format.

F401

Convert a standard calendar date (365/366 day year) to fiscal date format.

F402

Convert a fiscal date to a standard calendar date (365/366 day year).

C/P 403

Days between fiscal calendar dates.

C/P 404

Date +/-*n* days, weeks, periods, quarters, or years.

Each of the Fiscal Calendar functions is described in detail in later sections.

(F401) CALENDAR DATE to FISCAL DATE

This function will convert a date based on a standard calendar date (for example, Gregorian or Julian date) to a fiscal calendar date. Output dates are optional. The function will always store the following fiscal calendar date information in:

- TRC-CONVR-STDOUT-DATE-X (format CCYYPPDD)
  - CC—Fiscal Century
  - YY—Fiscal Year
  - PP—Fiscal Period
  - DD—Fiscal Day-of-Period
- TRC-CONVR-OUT-NUMERIC-PARM (format QQPPWWDD)
  - QQ—Fiscal Quarter number (01-04)
  - PP—Fiscal Period number (01-13)
  - WW—Fiscal Week number (01-52/53)
  - DD—Fiscal Day-of-Period number (01-37)

Function F401 converts a standard calendar date to a fiscal date definition defined in Alphanumeric Fiscal Date Definitions. See TRC-CONVR-From-Date-MASK for more information about standard calendar dates.

Alphanumeric FISCAL Date Definitions

In the definitions shown next, trailing dashes represent spaces.

- YY------YEAR (century omitted)
- CCYY----YEAR
- YYQQ----YEAR (century omitted) / Quarter
- CCYYQQ YEAR / Quarter
- YYPP----YEAR (century omitted) / Period
- CCYYPP--YEAR / Period
- YYPPDD--YEAR (century omitted) / Period / Day-of-Period

- CCYYPPDD YEAR / Period / Day-of-Period
- YYWW----YEAR (century omitted) / Week Number
- CCYYWW--YEAR / Week Number

Numeric Date Definitions

In the definitions shown next, leading dashes represent zeros.

- ------YY YEAR (century omitted)
- ----CCYY YEAR
- ----YYQQ YEAR (century omitted)/Quarter
- --CCYYQQ YEAR/Quarter
- ----YYPP YEAR (century omitted)/Period
- --CCYYPP YEAR/Period
- --YYPPDD YEAR (century omitted)/Period/Day-of-Period
- ----YYWW Year (century omitted)/Week Number
- --CCYYWW YEAR/Week Number

If a Fiscal Calendar Date mask is used, the input calendar date will always be converted to the Fiscal Calendar date based upon one of the previous date definitions (masks). If a non-Fiscal Calendar date mask is specified for output, the dates returned will contain values based on the INPUT-ALPHA-PARM:

Input-Alpha-Parm		Output-Date-Mask	Value
F-BOW	begin of week	CCYYMMDD	01-02-2000
B-EOW	end of week	CCYYMMDD	01-08-2000
F-BOP	begin of period	CCYYMMDD	01-02-2000
F-EOP	end of period	CCYYMMDD	01-29-2000
F-BOQ	begin of quarter	CCYYMMDD	01-02-2000
F-EOQ	end of quarter	CCYYMMDD	03-25-2000
F-BOY	begin of year	CCYYMMDD	01-02-2000
F-EOY	end of year	CCYYMMDD	12-30-2000

The following examples convert a standard calendar date to a 4-4-4 Fiscal Calendar date. The output date is optional.

Examples 1 and 2

CALENDAR DATE to FISCAL DATE Parameters	
Business	1. FISCAL-ID 2. From-Date-MASK 3. OUT1-DATE-MASK
Input	1. FUNCTION-CODE 2. CENTURY-BREAK 3. From-Date 4. IN-ALPHA-PARM
Output	1. STANDARD-DATE 2. Out-NUMERIC-PARM 3. OUT1-DATE

Example 2

Before Call	After Call
$444 YYMMDD-- CCYYMMDD	
F401 19/20;68 971231 F-BOW	
	19971332 +0413533 19971228

Example 3

Before Call	After Call
$444 ---YYDDD YYWW---	
F401 19/20;68 00097365 F-BOW	
	19971231 +04135332 9753

In the previous examples, the objective is to convert calendar date December 31, 1997 to Fiscal Calendar standard CA Calendar Routines output information. Example 1 contains '------' (nulls) in the Output 1 date mask field. Example 2 contains a valid Fiscal Calendar Date that generates the required output.

(F402) FISCAL DATE to CALENDAR DATE

This function converts a Fiscal Calendar date to a standard calendar date such as Fiscal $444 to Gregorian CCYYMMDD. Input to the function is a fiscal date in the From-Date field containing a date defined by one of the Fiscal Calendar date masks (From-Date-MASK).

If the input date contains a day value (mask contains DD—see the following table), the date is converted to an output date defined by the OUT1/3-DATE-MASK(S). If the input is missing the day value, the output date is set to the beginning or ending of a Fiscal Calendar week, period, quarter, or year based on the input From-Date-MASK and the INPUT-ALPHA parameter.

From-Date-Mask	Input-Alpha	Example	
CCYYPPDD	n/a	20070202	02-01-2007
YYPPDD-	n/a	07202	02-01-2007
--YYPPDD	n/a	00070202	02-01-2007
YY-----	F-BEGIN	07	01-03-2007
------YY	F-END	00000007	01-01-2008
CCYY---	F-BEGIN	2007	01-03-2007
----CCYY	F-END	00002007	01-01-2008
YYQQ---	F-BEGIN	0701	01-03-2007
----YYQQ	F-END	00000701	03-27-2007
CCYYQQ-	F-BEGIN	200701	01-03-2007
--CCYYQQ	F-END	00200701	03-27-2007
YYPP---	F-BEGIN	0701	01-03-2007
----YYPP	F-END	00000701	01-30-2007
CCYYPP-	F-BEGIN	200701	01-03-2007
--CCYYPP	F-END	00200701	01-30-2007
YYWW---	F-BEGIN	0701	01-03-2007
----YYWW	F-END	00000701	01-09-2007
CCYYWW-	F-BEGIN	200701	01-03-2007
--CCYYWW	F-END	00200701	01-09-2007

This function will convert a 4-4-4 Fiscal Calendar date to a standard calendar date (for example, Fiscal Date to Gregorian or Julian date). One Output date is required.

This function requires an input alphabetic parameter whenever a Special Fiscal Calendar date is being converted and the date does not contain a day value.

- F-BEGIN- beginning date
- F-END- ending date

FISCAL DATE to CALENDAR DATE Parameters	
Business	1. FISCAL-ID 2. From-Date-MASK 3. OUT1-DATE-MASK
Input	1. FUNCTION-CODE 2. CENTURY-BREAK 3. From-Date 4. IN-ALPHA-PARM
Output	1. STANDARD-DATE 2. OUT1-DATE

Example 1

Before Call	After Call
$444 YYPPDD--E-M-D-CY	
F402 19/20;68 971335 F-END	
	19980103 01-03-1998

Example 2

Before Call	After Call
$444 ---YYPP E-M/D/Y	
F402 19/20;68 00009902 F-END	
	19990227 02/27/99

The objective in the previous examples is to convert Fiscal $444 calendar dates to standard calendar dates. Example 2 contains a date mask and date that point to the beginning or end of a Fiscal 4-4-4 Calendar period. The F-END alpha parameter directs CA Calendar Routines to extract the ending date information from the Fiscal Calendar tables for the period specified in the From-Date field. All output from this function in OUT-1, OUT-2, and OUT-3 areas is formatted based on a Calendar Output Date Mask. The STANDARD-DATE field will always contain a Gregorian date.

(C/P 403) DAYS BETWEEN

This function will count the number of days from the From-Date to the To-Date. If the From-Date is earlier than the To-Date, the answer is positive. If the From-Date is later than the To-Date, the answer is negative. If the two dates are equal, then the answer is 0.

Examples 1 and 2

DAYS BETWEEN Parameters	
Business	1. HOLIDAY-TBL
	2. FISCAL-ID
	3. From-Date-MASK
	4. To-Date-MASK
	5. PROC-DAY-DEF
	6. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. To-Date
Output	1. OUT-NUMERIC-PARM

Example 1

Before Call	After Call
06	
$444	
CCYYPPDD	
CCYYPPDD	
'NEEEEEN'	
'T'	
C403	
20100903	
20100907	
	+4

Example 2	
Before Call	**After Call**
06	
$444	
CCYYPPDD	
CCYYPPDD	
'NEEEEEN'	
'T'	
P403	
20100903	
20100907	
	+1

Note: Period 9, day-of-period 6, 2010 is in Holiday Table 06 and it is a Monday. The FROM and TO dates are also accounted for in Fiscal Calendar table $444.

Our goal is to count the number of days from the From-Date to the To-Date. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In Example 1, every day qualifies; and in Example 2, only processing days qualify.

Example	Date	Qualification	Count
Example 1	20100903	N/A	+0
	20100904	YES	+1
	20100905	YES	+2
	20100906	YES	+3
	20100907	YES	+4
Example 2	20100903	N/A	+0
	20100904	NO, Saturdays are never processed.	+0
	20100905	NO, Sundays are never processed	+0
	20100906	NO, Monday-Holidays are not processed	+0
	20100907	YES	+1

Examples 3 and 4

DAYS BETWEEN Parameters	
Business	1. HOLIDAY-TBL
	2. FISCAL-ID
	3. From-Date-MASK
	4. To-Date-MASK
	5. PROC-DAY-DEF
	6. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. To-Date
Output	1. OUT-NUMERIC-PARM

Example 3

Before Call	After Call
06	
$454	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C403	
20100907	
20100903	
	-4

Example 4

Before Call	After Call
06	
$454	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	

Example 4	
Before Call	**After Call**
P403	
20100907	
20100903	
	-1

Note: Period 9, day-of-period 6, 2010 is in Holiday Table 06 and it is a Monday. The FROM and TO dates are also accounted for in Fiscal Calendar table $454.

Examples 3 and 4 are similar to Examples 1 and 2, except that the From-Date and the To-Date have been reversed.

Example	Date	Qualification	Count
Example 1	20100907	N/A	-0
	20100906	YES	-1
	20100905	YES	-2
	20100904	YES	-3
	20100903	YES	-4
Example 2	20100907	N/A	-0
	20100906	NO, Monday-Holidays are not processed.	-0
	20100905	NO, Sundays are never processed.	-0
	20100904	NO, Saturdays are never processed.	-0
	20100903	YES	-1

(C/P 404) DATE +/- N DAYS, WEEKS, PERIODS, QUARTERS, OR YEARS

This function will calculate the date that is *n* days, weeks, periods, quarters, or years later than or earlier than the specified Fiscal Calendar date.

This function requires an input alphabetic parameter specifying type of time period to increment or decrement.

Parameter	Explanation C/P 404 Fiscal Date +/-*n* Days, Weeks, Periods, Quarters, or Years
F-DAYS	increment/decrement n days
F-WEEKS	increment/decrement n weeks
F-PER	increment/decrement n periods
F-QTR	increment/decrement n quarters
F-YEARS	increment/decrement n years

Examples 1 and 2

DATE +/- N DAYS, WEEKS, PERIODS, QUARTERS, OR YEARS Parameters	
Business	1. HOLIDAY-TBL
	2. FISCAL-ID
	3. From-Date-MASK
	4. OUT1-DATE-MASK
	5. PROC-DAY-DEF
	6. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
	4. IN-ALPHA-PARM
Output	1. OUT1-DATE

Example 1

Before Call	After Call
06	
$444	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C404	
20100903	
+4	
F-DAYS	
	20100907

Example 2

Before Call	After Call
06	
$444	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P404	
20100903	
+4	
F-DAYS	
	20100910

Note: Period 9, day-of-period 6, 2010 is in Holiday Table 06 and it is a Monday. The FROM and TO dates are also accounted for in Fiscal Calendar table $444.

Our goal is to calculate the date that is +4 days from the From-Date. According to the meaning of END-PNTS-DEF, the From-Date corresponds to day count 0. In example 1, every day qualifies; and in example 2, only processing days qualify

Example	Date	Qualification	Count
Example 1	20100903	N/A	+0
	20100904	YES	+1
	20100905	YES	+2
	20100906	YES	+3
	20100907	YES	+4
Example 2	20100903	N/A	+0
	20100904	NO, Saturdays are never processed.	+0
	20100905	NO, Sundays are never processed	+0
	20100906	NO, Monday-Holidays are not processed	+0
	20100907	YES	+1
	20100908	YES	+2
	20100909	YES	+3
	20100910	YES	+4

Examples 3 and 4

DATE +/- N DAYS, WEEKS, PERIODS, QUARTERS, OR YEARS Parameters	
Business	1. HOLIDAY-TBL
	2. FISCAL-ID
	3. From-Date-MASK
	4. OUT1-DATE-MASK
	5. PROC-DAY-DEF
	6. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
	4. IN-ALPHA-PARM
Output	1. OUT1-DATE

Example 3

Before Call	After Call
06	
$454	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C404	
20100907	
-4	
F-DAYS	
	20100903

Example 4

Before Call	After Call
06	
$454	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P404	
20100910	
-4	
F-DAYS	
	20100903

Note: Period 9, day-of-period 6, 2010 is in Holiday Table 06 and it is a Monday. The FROM and TO dates are also accounted for in Fiscal Calendar table $454.

Examples 3 and 4 are similar to examples 1 and 2, except that the direction is toward the past, rather than toward the future. (–4 rather than +4)

Example	Date	Qualification	Count
Example 1	20100907	N/A	-0
	20100906	YES	-1
	20100905	YES	-2
	20100904	YES	-3
	20100903	YES	-4
Example 2	20100910	N/A	-0
	20100909	YES	-1
	20100908	YES	-2
	20100907	YES	-3
	20100906	NO, Monday-Holidays are not processed.	-3
	20100905	NO, Sundays are never processed.	-3
	20100904	NO, Saturdays are never processed	-3
	20100903	YES	-4

ADVANCED Functions

There are 288 ADVANCED functions that perform specialized tasks. They answer questions such as “What date is three days prior to the end of the previous fiscal quarter?”

Function Type

The Function Type for every ADVANCED function is either C or P.

- C means that CA Calendar Routines considers all calendar days.
- P means that CA Calendar Routines considers only processing days.

Function Number

The Function Number for advanced functions has a format of *Fxx* where F ranges from 5 through 8, and xx ranges from 1 through 36.

Type of Function (F)

The value of F (5, 6, 7 or 8) refers to the type of Function:

5xx = Date Identification

500 Series

1 Identify past (or future) date

6xx = Days Since/Until

600 Series

1 Identify past (or future) date

2 Calculate days since (or until)

7xx = Date Relative To

700 Series

1 Identify past (or future) date

2 Calculate date “relative to”

8xx = Nth DOW Within a Period

800 Series

1 Identify past (or future) date

2 Calculate nth DOW occurrence from the beginning or end of the period

Time-Period of Interest (xx)

The value of *xx* (01-36) refers to the time-period of interest.

Time-Period			FUNC
Calendar Year	Previous Previous	Beginning	F01
		End	F02
	Current Current	Beginning	F03
		End	F04
	Next Next	Beginning	F05
		End	F06

Time-Period			FUNC
Fiscal Year	Previous	Beginning	F07
	Previous	End	F08
	Current	Beginning	F09
	Current	End	F10
	Next	Beginning	F11
	Next	End	F12
Calendar Quarter	Previous	Beginning	F13
	Previous	End	F14
	Current	Beginning	F15
	Current	End	F16
	Next	Beginning	F17
	Next	End	F18
Fiscal Quarter	Previous	Beginning	F19
	Previous	End	F20
	Current	Beginning	F21
	Current	End	F22
	Next	Beginning	F23
	Next	End	F24
Calendar Month	Previous	Beginning	F25
	Previous	End	F26
	Current	Beginning	F27
	Current	End	F28
	Next	Beginning	F29
	Next	End	F30
Fiscal Month	Previous	Beginning	F31
	Previous	End	F32
	Current	Beginning	F33
	Current	End	F34
	Next	Beginning	F35
	Next	End	F36

Date Identification (5xx)

The Date Identification functions provide a date of particular interest.

Examples C51 and P51

Date Identification Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. FISCYR-START
	5. PROC-DAY-DEF
	6. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
Output	1. OUT1-DATE

Example C51	
Before Call	**Before Call**
08	
CCYYMMDD	
CCYYMMDD	
04	
'NEEEEEN'	
'T'	
C519	
CURRDATE	
	20100101

Example P51	
Before Call	**After Call**
08	
CCYYMMDD	
CCYYMMDD	
04	
'NEEEEEN'	
'T'	
P519	
CURRDATE	
	20100104

Note: The current date is May 17, 2010. Holiday Table 08 contains January 01, 2010 and it is a Friday.

- Example C51. Our goal is to identify the calendar beginning of the previous fiscal quarter. Since the fiscal year starts April 1, the reference date, May 17, 2010, lies in the first fiscal quarter. The beginning of the previous fiscal quarter is January 01, 2010. Since we have requested the calendar beginning, we have our answer.
- Example P51. This is the same as the example C51, except that we have requested the processing beginning of the previous fiscal quarter. Since January 01, 2010 is a holiday and it is a Friday, it is not considered a processing day. Therefore we must perform a forward adjustment. We do this by searching toward the future until we encounter a processing day. Saturday and Sunday are never processed, so we do not find a processing day until January 04, 2010. That date becomes our answer.

Examples C52 and P52

Date Identification Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
Output	1. OUT1-DATE

Example C52

Before Call	After Call
05	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C504	
20101129	
	20101231

Example P52

Before Call	After Call
05	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P504	
20101129	
	20101230

Note: Holiday Table 05 contains December 31, 2010 and it is a Friday.

- Example C52. Our goal is to identify the calendar end of the calendar year containing November 29, 2010. The calendar end of 2010 is December 31, 2010, which is our answer.
- Example P52. This is the same as example C52, except that we have requested the processing end of the calendar year containing November 29, 2010. Since December 31, 2010 is a holiday and it is a Friday, it is not considered a processing day. Therefore, we must perform a backward adjustment. We do this by searching toward the past until we encounter a processing day. December 30, 2010 qualifies, and so we have our answer.

Days Since/Until (6xx)

The Days Since/Until functions provide a number that specifies the direction and displacement from the From-Date to the calendar date of particular interest.

Examples C61 and P61

Days Since/Until Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. FISCYR-START 4. PROC-DAY-DEF 5. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date
Output	1. OUT-NUMERIC-PARM

Example C61

Before Call	After Call
08 CCYYMMDD 04 'NEEEEEN' 'T'	
C619 CURRDATE	
	-136

Example P61

Before Call	After Call
08 CCYYMMDD 04 'NEEEEEN' 'T'	
P619 CURRDATE	
	-95

Note: The current date is May 17, 2010. Holiday Table 08 contains Jan 01, 2010 and it is a Friday.

- Example C61. Our goal is to calculate the direction and displacement, in calendar days, from the current date to the calendar beginning of the previous fiscal quarter. Since the fiscal year starts April 1, the reference date, May 17, 2010, lies in the first quarter. The beginning of the previous fiscal quarter is January 01, 2010. Since we have requested the calendar beginning, January 01, 2010 becomes the To-Date. CA Calendar Routines will then calculate the direction and displacement, in calendar days, from May 17, 2010 to January 01, 2010. The answer is -136 calendar days.
- Example P61. This is the same as example C61, except that we have requested the direction and displacement, in processing days, from the current date to the processing beginning of the previous fiscal quarter. Since January 01, 2010 is a holiday and it is a Friday, it is not considered a processing day. Therefore we must perform a forward adjustment. We do this by searching toward the future until we encounter a processing day. Saturday and Sunday are never processed, so we do not find a processing day until January 04, 2010. That date becomes the To-Date. CA Calendar Routines will then calculate the direction and displacement, in processing days from May 17, 2010 to January 04, 2010. The answer is -95 processing days.

Examples C62 and P62

Days Since/Until Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. PROC-DAY-DEF
	4. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
Output	1. OUT-NUMERIC-PARM

Example C62

Before Call	After Call
05	
CCYYMMDD	
'NEEEEEN'	
'T'	
C604	
20101129	
	+32

Example P62	
Before Call	**After Call**
05	
CCYYMMDD	
'NEEEEEN'	
'T'	
P604	
20101129	
	+22

Note: Holiday Table 05 contains December 31, 2010 and it is a Friday. Holiday Table 05 also contains December 24, 2010 and it is a Friday.

- Example C62. Our goal is to identify the direction and displacement, in calendar days, from November 29, 2010 to the calendar end of the calendar year containing November29, 2010. The calendar end of 2010 is December 31, 2010, which becomes the To-Date. CA Calendar Routines will then calculate the direction and displacement, in calendar days, from November 29, 2010 to December 31, 2010. The answer is +32 calendar days.
- Example P62. This is the same as example C62, except that we have requested the direction and displacement, in processing days from November 29, 2010 to the processing end of the calendar year, containing November 29, 2010. Since December 31, 2010 is a holiday and it is a Friday, it is not considered a processing day. Therefore, we must perform a backward adjustment. We do this by searching toward the past until we encounter a processing day. December 30, 2010 qualifies, so it becomes the To-Date. CA Calendar Routines will then calculate the direction and displacement, in processing days, from November 29, 2010 to December 30, 2010. The answer is +22 processing days.

DATE RELATIVE TO (7xx)

These functions provide a date that is determined by starting at a calendar date of particular interest, and then advancing forward or backward a specified number of days.

Examples C71 and P71

DATE RELATIVE TO Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. FISCYR-START
	5. PROC-DAY-DEF
	6. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example C71

Before Call	After Call
08	
CCYYMMDD	
CCYYMMDD	
04	
'NEEEEEN'	
'T'	
C719	
CURRDATE	
+5	
	20100106

Example P71

Before Call	After Call
08	
CCYYMMDD	
CCYYMMDD	
04	
'NEEEEEN'	
'T'	
P719	
CURRDATE +5	
	20100111

Note: The current date is May 17, 2010. Holiday Table 08 contains Jan 01, 2010 and it is a Friday.

- Example C71. Our goal is to calculate the date that is +5 calendar days from the calendar beginning of the previous fiscal quarter. Since the fiscal year starts April 1, the reference date, May 17, 2010, lies in the first fiscal quarter. The beginning of the previous fiscal quarter is January 01, 2010. Since we have requested the calendar beginning, the date January 01, 2010 becomes the new From-Date. CA Calendar Routines will then calculate the date that is +5 calendar days from the new From-Date, January 01, 2010. The answer is January 06, 2010.
- Example P71. This is the same as the example C71, except that we have requested the date that is +5 processing days from the processing beginning of the previous fiscal quarter. Since January 01, 2010 is a holiday and it is a Friday, it is not considered a processing day. Therefore we must perform a forward adjustment. We do this by searching toward the future until we encounter a processing day. Saturday and Sunday are never processed, so we do not find a processing day until January 04, 2010. That date becomes the new From-Date. CA Calendar Routines will then calculate the date that is +5 processing days from the new From-Date, January 04, 2010. The answer is January 11, 2010.

Examples C72 and P72

DATE RELATIVE TO Parameters	
Business	1. HOLIDAY-TBL
	2. From-Date-MASK
	3. OUT1-DATE-MASK
	4. PROC-DAY-DEF
	5. END-PNTS-DEF
Input	1. FUNCTION-CODE
	2. From-Date
	3. IN-NUMERIC-PARM
Output	1. OUT1-DATE

Example C72

Before Call	After Call
05	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
C704	
20101129 -6	
	20101225

Example P72

Before Call	After Call
05	
CCYYMMDD	
CCYYMMDD	
'NEEEEEN'	
'T'	
P704	
20101129	
-6	
	20101221

Note: Holiday Table 05 contains December 31, 2010 and it is a Friday. Holiday Table 05 also contains December 24, 2010 and it is a Friday.

- Example C72. Our goal is to identify the date that is -6 calendar days from the calendar end of the calendar year containing November 29, 2010. The calendar end of 2010 is December 31, 2010, which becomes the new From-Date. CA Calendar Routines will then calculate the date that is -6 calendar days from the new From-Date, December 31, 2010. The answer is December 25, 2010.
- Example P72. This is the same as the example C72, except that we have requested the date that is -6 processing days from the processing end of the calendar year, containing November 29, 2010. Since December 31, 2010 is a holiday and it is a Friday, it is not considered a processing day. Therefore, we must perform a backward adjustment. We do this by searching toward the past until we encounter a processing day. December 30, 2010 qualifies, so it becomes the new From-Date. CA Calendar Routines will then calculate the date that is -6 processing days from the new From-Date, December 30, 2010. This will yield the answer we seek, December 21, 2010.

Nth DOW OCCURRENCE WITHIN A PERIOD (8xx)

These functions are executed in three stages.

1. The period beginning or end you want is determined relative to the From-Date.
2. CA Calendar Routines will count forward from the Period-Beginning (for odd functions) or backward from the Period-End (for even functions) tallying only the wanted days of the week.
3. CA Calendar Routines makes sure that the answer is still within the Period.

Examples C81 and P81

Nth DOW OCCURRENCE WITHIN A PERIOD Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. OUT1-DATE-MASK 4. PROC-DAY-DEF 5. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM 4. IN-ALPHA-PARM
Output	1. RETURN-NUM 2. OUT1-DATE

Example C81

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C827 20091215 +4 'NNNNNYN'	
	000 20091225

Example P81

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
P827 20091215 +4 'NNNNNYN'	
	054

Note: December 25, 2009 is in Holiday Table 08 and it is a Friday.

- Example C81. Our goal is to calculate the date that is the fourth calendar Friday after the calendar beginning of the calendar month specified by December 15, 2009. The calendar beginning of the calendar month containing December 15, 2009 is December 01, 2009. CA Calendar Routines then counts forward four Calendar Fridays until it reaches December 25, 2009, which becomes the answer.
- Example P81. This is the same as the example C81, except that we have requested the date that is the fourth processing Friday after the processing beginning of the calendar month specified by December 15, 2009. The analysis starts out, as before, with December 01, 2009. However this time, when CA Calendar Routines counts ahead four processing Fridays, it must skip December 25, 2009, because it is not a processing day. The actual answer therefore lies in January, which results in error message 054, because the wanted answer is beyond the end of the specified period.

Examples C82 and P82

Nth DOW OCCURRENCE WITHIN A PERIOD Parameters	
Business	1. HOLIDAY-TBL 2. From-Date-MASK 3. OUT1-DATE-MASK 4. PROC-DAY-DEF 5. END-PNTS-DEF
Input	1. FUNCTION-CODE 2. From-Date 3. IN-NUMERIC-PARM 4. IN-ALPHA-PARM
Output	1. RETURN-NUM 2. OUT1-DATE

Example C82

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
C828 20091215 -4 'NNNNNYN'	
	000 20091204

Example P82

Before Call	After Call
08 CCYYMMDD CCYYMMDD 'NEEEEEN' 'T'	
P828 20091215 -4 'NNNNNYN'	
	053

Note: December 25, 2009 is in Holiday Table 08 and it is a Friday.

- Example C82. Our goal is to calculate the date that is the fourth calendar Friday prior to the calendar end of the calendar month specified by December 15, 2009. The calendar end of the calendar month containing December 15, 2009 is December 31, 2009. CA Calendar Routines then counts backward four calendar Fridays until it reaches December 04, 2009, which becomes the answer.
- Example P82. This is the same as the example C82, except that we have requested the date that is the fourth processing Friday prior to the processing end of the calendar month specified by December 15, 2009. The analysis starts out, as before, with December 31, 2009. However this time, when CA Calendar Routines counts backwards four processing Fridays, it must skip December 25, 2009, because it is not a processing day. The actual answer therefore lies in November, which results in error message 053, because the wanted answer is earlier than the beginning of the specified period.

Appendix A: Return Codes

This appendix gives a brief description of the return codes that are generated by CA Calendar Routines.

The chapter includes the following sections:

- Overview
- Good return
- Bad input data
- Bad DATE-MASK Messages
- Execution time errors
- Systems problems

Overview

The Return Code contains a three digit number (Return Number) and eighty characters of associated text (Return Text). CA Calendar Routines will set the Return Number to a value that falls into one of the categories shown below.

Category	Range
Good Return	000
Bad Input Data	001-019, 020-029, 030-039, 040-045
Execution Time Errors	051-054, 099
Systems Problems	900-999

Each of these categories is described in greater detail in their respective sections.

Good Return

This section describes message 000.

000

GOOD RETURN

Reason:

The function executed successfully.

Action:

None required.

Bad Input Data

This section describes messages 001-017, 030-039, and 040-042.

001

BAD HOLIDAY-TBL COULD NOT BE ACCESSED

Reason:

You specified a Holiday Table that is not known to your system. It must be 00 through ZZ, defined by your administrator, and available for access.

Action:

If you have entered a value of 00 through ZZ, talk to your administrator to see why the table is not known or available.

Note: This is an edit message if the tables are hardcoded in working storage as a COBOL copybook, otherwise it is a run time message stating the holiday table could not be accessed as a record or a dynamically called program. Your system administrator will be able to interpret the message for you.

002

BAD FISCYR-START

Reason:

You specified a Fiscal Start month that is not 01 through 12.

Action:

You must specify a value of 01 through 12. This represents the first month of the Fiscal Year. If the fiscal year coincides with the Calendar Year, then you should specify the value 01.

003

BAD FISCMO-START

Reason:

You specified a Fiscal Start day that is not 01 through 28.

Action:

You must specify a value of 01 through 28. This represents the first day of the Fiscal Month. If the fiscal month coincides with the Calendar month, then you should specify the value 01.

004

BAD END-PNTS-DEF

Reason:

You specified a value for the End Points Definition that is not B, N, F, or T.

Action:

You must specify B, N, F, or T, for the description of the End Points Definition.

005

BAD DOW-STRING

Reason:

You specified a value that is not numeric or that one of the valid numerals has already been specified.

Action:

Each value that you specify for Sunday through Saturday must be numeric and must not repeat itself.

006

BAD PROC-DAY-DEF

Reason:

You specified a processing day definition that is not A, N, E or I.

Action:

Each value that you specify for Sunday through Saturday, must describe the processing day as A, N, E, or I.

007

BAD CENTURY-BREAK

Reason:

You specified a century or year that is not permitted.

Action:

Both centuries and the year must be numeric. Additionally, neither century is allowed to be zero, and finally, the lower century must be one less than the upper century.

008

BAD FROM-DATE-MASK

Reason:

See Bad DATE-MASK Messages.

Action:

Specify one of the permitted date masks.

009

BAD TO-DATE-MASK

Reason:

See Bad DATE-MASK Messages.

Action:

Specify one of the permitted date masks.

010

BAD OUT1-DATE-MASK

Reason:

See Bad DATE-MASK Messages.

Action:

Specify one of the permitted date masks.

011

BAD OUT2-DATE-MASK

Reason:

See Bad DATE-MASK Messages.

Action:

Specify one of the permitted date masks.

012

BAD OUT3-DATE-MASK

Reason:

See Bad DATE-MASK Messages.

Action:

Specify one of the permitted date masks.

Bad DATE-MASK Messages

Messages 008 to 012 mean that you have specified a date mask that is not one of the permitted categories. There are several categories of masks available. See the Business Parameters section of the “Communications” chapter for details.

Using Date Elements

You can describe the arrangement of various pieces of date information, such as year, month and day. There are five such elements: CC century, YY year, MM Gregorian Month, DD Gregorian Day, and DDD Julian Day. The character - corresponds to leading zeroes (numeric) or trailing spaces (alphanumeric). (Use FROM-DATE-X or FROM-DATE-9 as the input area).

```
"---YYDDD" Numeric Move of YYDDD"YYMMDD--" Alphanumeric Move of YYMMDD
```

Single Century Designation

You can also use the following masks that will let you designate a century as a single character.

- If H is 8, the century is 18.
- If H is 9, the century is 19.
- If H is 0, the century is 20.
- If R is 0 the century is 19.
- If R is 1, the century is 20.
- If S is 1 the century is 19.
- If S is 2, the century is 20.
- If T is 0, the century is 18.
- If T is 1, the century is 19.
- If T is 2, the century is 20.

The character "-" corresponds to leading zeroes (numeric) or trailing spaces (alphanumeric). (Use FROM-DATE-X or FROM-DATE-9 as the input area.)

```
"-HYYMMDD" "-HMMDDYY" "HYYMMDD-" "HMMDDYY-"
"--HYYDDD" "--HDDDYY" "HYYDDD--" "HDDDYY--"
"-RYYMMDD" "-RMMDDYY" "RYYMMDD-" "RMMDDYY-"
"--RYYDDD" "--RDDDYY" "RYYDDD--" "RDDDYY--"
"-SYYMMDD" "-SMMDDYY" "SYYMMDD-" "SMMDDYY-"
"--SYYDDD" "--SDDDYY" "SYYDDD--" "SDDDYY--"
"-TYYMMDD" "-TMMDDYY" "TYYMMDD-" "TMMDDYY-"
"--TYYDDD" "--TDDDYY" "TYYDDD--" "TDDDYY--"
```

360-Day Calendar Designation

You can also use the following masks to designate days within a 360-DAY Calendar Year. These masks are only allowed in the K300 series of functions, and KKK represents a value from 001 through 360. The character "-" corresponds to leading zeroes (numeric) or trailing spaces (alphanumeric). (Use FROM-DATE-X or FROM-DATE-9 as the input area).

```
"CCYYKKK-" "-CCYYKKK" "KKKCCYY-" "-KKKCCYY"
"YYKKK---" "---YYKKK" "KKKYY---" "---KKKYY"
```

Edited Masks

You can also use the following edited masks. (Use FROM-DATE-E as the input area.)

Normal Edited

```
"E-M/D/Y " equals MM/DD/YY
"E-Y/M/D " equals YY/MM/DD
"E-Y-M-D " equals YY-MM-DD
"E-M-D-Y " equals MM-DD-YY
"E-D-M-Y " equals DD-MM-YY
"E-CY-M-D" equals CCYY-MM-DD
"E-M-D-CY" equals MM-DD-CCYY
"E-D-M-CY" equals DD-MM-CCYY
"E-YY-DDD" equals YY-DDD
"E-M D Y " equals MM DD YY
"E-M D CY" equals MM DD CCYY
"E-DMONY " equals DDmmmYY e.g. 20FEB95
"E-DMONCY" equals DDmmmCCYY e.g. 20FEB1995
```

Note: “E-DMONY” and “E-DMONCY” are also included as “E-TDS-NN” formats. This format supports the non-English versions of these date masks. These date masks will continue to be supported in the English language versions.

Special DB2 Formats

```
"E-DB2ISO" equals CCYY-MM-DD
"E-DB2USA" equals MM/DD/CCYY
"E-DB2EUR" equals DD.MM.CCYY
"E-DB2JIS" equals CCYY-MM-DD
```

Special AS400 Formats

```
"E-A4*ISO" equals CCYY-MM-DD
"E-A4*USA" equals MM/DD/CCYY
"E-A4*EUR" equals DD.MM.CCYY
"E-A4*JIS" equals CCYY-MM-DD
"E-A4*MDY" equals MM/DD/YY
"E-A4*DMY" equals DD/MM/YY
"E-A4*YMD" equals YY/MM/DD
"E-A4*JUL" equals YY/DDD
```

User Requested

(Notify CA if you would like a format that is not listed).

```
"E-TDS-01"  equals DD mmm YY (18 DEC 95)
"E-TDS-02"  equals DD-mmm-YY (18-DEC-95)
"E-TDS-03"  equals DD/mmm/YY (18/DEC/95)
"E-TDS-04"  equals DD mmm CCYY  (18 DEC 1995)
"E-TDS-05"  equals DD-mmm-CCYY  (18-DEC-1995)
"E-TDS-06"  equals DD/mmm/CCYY  (18/DEC/1995)
"E-TDS-07"  equals mmm. DD, CCYY (DEC. 18, 1995)
"E-TDS-08"  equals mmm DD CCYY  (DEC 18 1995)
"E-TDS-09"  equals mmm-DD-CCYY  (DEC-18-1995)
"E-TDS-10"  equals mmm/DD/CCYY  (DEC/18/1995)
"E-TDS-11"  equals MMMMM DD, CCYY  (DECEMBER 18, 1995)
"E-TDS-12"  equals DD.MM.YY (18.12.95)
"E-TDS-13"  equals CCYY MMM DD  (1995 DEC 18)
"E-TDS-14"  equals CCYY/MM/DD  (1995/12/18)
"E-TDS-15"  equals MM/DD/YYCC  (18/12/9519)
"E-TDS-16"  equals DDmmmYY  (18DEC95)
"E-TDS-17"  equals DDmmmCCYY (18DEC1995)
"E-TDS-18"  equals ddd DD mmm YY (MON 18 DEC 95)
"E-TDS-19"  equals ddd DD-mmm-YY (MON 18-DEC-95)
"E-TDS-20"  equals ddd DD/mmm/YY (MON 18/DEC/95)
"E-TDS-21"  equals ddd DD mmm CCYY (MON 18 DEC 1995)
"E-TDS-22"  equals ddd DD-mmm-CCYY (MON 18-DEC-1995)
"E-TDS-23"  equals ddd DD/mmm/CCYY (MON 18/DEC/1995)
"E-TDS-24"  equals ddd mmm. DD. CCYY  (MON DEC. 18. 1995)
"E-TDS-25"  equals ddd mmm DD CCYY (MON DEC 18 1995)
"E-TDS-26"  equals ddd mmm-DD-CCYY (MON DEC-18-1995)
"E-TDS-27"  equals ddd mmm/DD/CCYY (MON DEC/18/1995)
"E-TDS-28"  equals ddd MMMMM DD, CCYY (MON DECEMBER 18, 1995)
"E-TDS-29"  equals ddd DD.MM.YY (MON 18.12.95)
"E-TDS-30"  equals ddd CCYY mmm DD (MON 1995 DEC 18)
"E-TDS-31"  equals ddd CCYY/MM/DD  (MON 1995/12/18)
"E-TDS-32"  equals ddd MM/DD/YYCC  (MON 12/18/9519)
"E-TDS-33"  equals ddd DDmmmYY  (MON 18DEC95)
"E-TDS-34"  equals ddd DDmmmCCYY (MON 18DEC1995)
"E-TDS-35" equals DDDDD DD mmm YY (MONDAY 18 DEC 95)
"E-TDS-36"  equals DDDDD DD-mmm-YY (MONDAY 18-DEC-95)
"E-TDS-37"  equals DDDDD DD/mmm/YY (MONDAY 18/DEC/95)
"E-TDS-38" equals DDDDD DD mmm CCYY (MONDAY 18 DEC 1995)
"E-TDS-39"  equals DDDDD DD-mmm-CCYY  (MONDAY 18-DEC-1995)
"E-TDS-40"  equals DDDDD DD/mmm/CCYY  (MONDAY 18/DEC/1995)
"E-TDS-41"  equals DDDDD mmm. DD. CCYY (MONDAY DEC. 18. 1995)
"E-TDS-42"  equals DDDDD mmm DD CCYY  (MONDAY DEC 18 1995)
"E-TDS-43"  equals DDDDD mmm-DD-CCYY  (MONDAY DEC-18-1995)
"E-TDS-44"  equals DDDDD mmm/DD/CCYY  (MONDAY DEC/18/1995)
"E-TDS-45"  equals DDDDD MMMMM DD, CCYY (MONDAY DECEMBER 18,1995)
"E-TDS-46"  equals DDDDD mmm DD.MM.YY (MONDAY 18.12.95)
```

```
"E-TDS-47"  equals DDDDD CCYY mmm DD (MONDAY 1995 DEC 18)
"E-TDS-48"  equals DDDDD CCYY/MM/DD (MONDAY 1995/12/18)
"E-TDS-49"  equals DDDDD MM/DD/YYCC (MONDAY 12/18/9519)
"E-TDS-50"  equals DDDDD DDmmmYY (MONDAY 18DEC95)
"E-TDS-51"  equals DDDDD DDmmmCCYY (MONDAY 18DEC1995)
"E-TDS-52"  equals MMMMM CCYY  (DECEMBER 1995)
"E-TDS-53"  equals mmm DD, CCYY (DEC 18, 1995)
"E-TDS-54"  equals YY.DD (99.351)
"E-TDS-55"  equals CCYY.DDD (1999.351)
"E-TDS-56"  equals mmm.DD, CCYY (DEC.18, 1995)
"E-TDS-57"  equals ZM-ZD-ZY (2-18-95)
"E-TDS-58"  equals ZM/ZD/ZY (2/18/
```

User Requested – Non-English Date Masks

Date masks identical in format to the above English E-TDS-nn date masks are provided in the following languages:

IBM-Country Code	Country
AT	Austria
DK	Denmark
FI	Finland
FR	France
NL	Netherlands
IT	Italy
DE	Germany
NO	Norway
SE	Sweden
ES	Spain

To obtain a non-English User Requested date format, modify the Edited Date mask as follows:

```
E-TDS-nn ETDSnnxx xx = above country code
E-TDS-45 (English) ETDS45ES (Spanish)
MONDAY DECEMBER 18, 1995 LUNES DICIEMBRE 18, 1995
```

Special Null Mask

Special Null Mask for use in Output Fields if multiple output date formats are not desired. (May improve execution performance.)

```
"--------" equals null mask (i.e. no output produced)
```

9-S Complement Masks

You can also use the following 9-S complement masks. These dates represent the original date for which each digit has been subtracted from 9. (Use FROM-DATE-X or FROM-DATE-9 as the input area).

```
"9-CYMD " equals 9's compl of CCYYMMDD
"9-N-YMD " equals 9's compl of YYMMDD PIC 9
"9-N-MDY " equals 9's compl of MMDDYY PIC 9
"9-N-CYJ " equals 9's compl of CCYYDDD PIC 9
"9-N-YJ " equals 9's compl of YYDDD PIC 9
"9-A-YMD " equals 9's compl of YYMMDD PIC X
"9-A-MDY " equals 9's compl of MMDDYY PIC X
"9-A-CYJ " equals 9's compl of CCYYDDD PIC X
"9-A-YJ " equals 9's compl of YYDDD PIC X
```

SERIAL-DATE Masks

You can also use the following SERIAL-DATE masks. These dates represent a displacement from a predefined reference date. Several such conventions exist for you to choose from. TDS is the CA Standard, S-1900G follows the 100 leap year rule, S-1900NG does not. (Use FROM-DATE-S as the input area).

```
Convention  Low High Reference DayCount
S-TDS 1 3652056 Jan 01, 0001 1
S-DB2 1 3652059 Jan 01, 0001 1
S-LE370 -577734 3074324 Oct 15, 1582 1
S-COB370 -584387 3067671 Jan 01, 1601 1
S-ANSI89 -584387 3067671 Jan 01, 1601 1
S-CICS-A -693594 2958464 Jan 01, 1900 1
S-1900G -693594 2958464 Jan 01, 1900 1
S-1900NG -693594 2958468 Jan 01, 1900 1
```

User-defined SERIAL DATE masks *(Snnnnnnn)* let you define the low and high references shown previously. The format for this date mask is shown in the following example:

```
Snnnnnnn User serial date based on date value supplied in "nnnnnnn".
Digits 6-7 = Day Value (e.g., 15)
Digits 4-5 = Month Value (e.g., 12)
Digits 2-3 = Year Value (e.g., 97)
Digit 1 designates the century number, where:0 for 2000 5 for 15001 for 2100 6
for 16002 for 2200 7 for 17003 for 2300 8 for 18004 for 2400 9 for 1900
Serial date mask identifier (constant)
```

013

BAD FUNCTION CODE

Reason:

You specified an invalid Function Code.

Action:

- Function Type—You must specify U, C, P, F, or K.
- Function Number—For U, you must specify 001 through 013. For C or P you must specify 101-103, 201-207, 403-404, 501-536, 601-636, 701-736, or 801-836. For K, you must specify 301 through 304. For F, you must specify 401 or 402.

014

BAD FROM-DATE

Reason:

The FROM-DATE is not valid.

Action:

The action depends on the FROM-DATE-MASK. See the appropriate FROM-DATE-MASK and use these editing criteria.

- Numeric needs leading zeros; alphanumeric needs trailing spaces.
- Gregorian:
  - CCYY—0001 through 9999
  - YY—00 through 99
  - MM—01 through 12
  - DD—1 through max days per month
- Julian:
  - CCYY—0001 through 9999
  - YY—00 through 99
  - DDD—001 through 365/366
- Single-Digit Century Designation: H must be 8, 9, or 0. R must be 0 or 1. S must be 1 or 2. T must be 0, 1, or2.

- Calendar-360 Dates: KKK must be 001 through 360.
- Edited Dates: Must correspond to the mask and be left justified.
- 9's Complement Dates: Must be a valid date when subtracted from all 9's.
- Serial Dates must be numeric, within range and not correspond to February 29 of a non-leap year.

015

BAD-TO-DATE

Reason:

The TO-DATE is not valid.

Action:

The action depends on the TO-DATE-MASK. For a detailed description of the interaction between the Mask and the Date, see the previous error message BAD FROM-DATE.

016

BAD IN-NUMERIC-PARM

Reason:

You specified a value in IN-NUMERIC-PARM that is not valid.

Action:

You must supply a numeric value in IN-NUMERIC-PARM. Also:

- Function 207—IN-NUMERIC-PARM must not equal 0.
- Odd Functions 801-836—IN-NUMERIC-PARM must be greater than 0.
- Even Functions 801-836—IN-NUMERIC-PARM must be less than 0.

017

BAD IN-ALPHA-PARM

Reason:

You specified an invalid value in this parameter.

Action:

The action varies depending on the function.

- Function 207 or 801-836—You must supply only Y or N in IN-ALPHA-PARM, for each day of the week.
- Function 103—You must supply one of the following strings: WEEKS, M-ENF, M-NOENF, Y-ENF, or Y-NOENF.
- Function 304—You must supply one of the following strings: K360-DY, K360-MO, or K360-YR.
- Function 010—You must supply one of the following strings:
  - ENFORCE
  - NO-ENF

018

BAD NON-ENGLISH USER DATE MASK

Reason:

The format for Non-English User Defined date mask is *ETDSnnll*, where *nn* must be 01-52 and *ll* must be one of the two-character language codes listed in the previous section, User Requested – Non-English Date Masks.

Action:

Revise the mask to comply with the Function, Language Code, and date mask.

019

CANNOT DETERMINE $NNN YEAR BEGINNING -OR-QUARTER BEGINNING -OR-PERIOD BEGINNING -OR- WEEK BEGINNING

Reason:

The Special Fiscal Calendar date input could not be located in the specified ($NNN) Fiscal Calendar table.

Action:

Correct the Fiscal Calendar date input (FROM-DATE) or change the Fiscal Calendar table ID.

020

BAD FISCAL-TABLE COULD NOT BE ACCESSED.

Reason:

The Special Fiscal Calendar table identification specified in TRC-FISCAL-ID is not defined in CA Calendar Routines.

Action:

Revise the four-character ID to match the Fiscal Calendar defined in CA Calendar Routines.

021

FISCAL CALENDAR ID INVALID, MUST BE $4NN.

Reason:

Special Fiscal Calendar IDs must be $444, $445, $454, or $455.

Action:

Revise the ID to match one of the valid values.

022

BAD (FISCAL) IN-ALPHA-PARM, SEE MANUAL.

Reason:

The IN-ALPHA-PARM specified for a Special Fiscal Calendar function does not correspond to a value specific to that function.

Action:

Revise the IN_ALPHA_PARM to a value specified for the function (see SPECIAL FISCAL CALENDAR Functions).

023

BAD (FISCAL) FROM-DATE-MASK. SEE MANUAL.

Reason:

The from date mask requires a Special Fiscal Calendar mask.

Action:

Specify one of the permitted Fiscal Calendar date masks.

024

INVALID FISCAL DATE INPUT

Reason:

A Special Fiscal Input date does not correspond to the FROM or TO INPUT-DATE_MASK.

Action:

The input date must match one of the date formats shown next.

Alphanumeric FISCAL date definitions:

YY------YEAR (century omitted)

CCYY----YEAR

YYQQ----YEAR (century omitted)/Quarter

CCYYQQ YEAR/Quarter

YYPP----YEAR (century omitted)/Period

CCYYPP--YEAR/Period

YYPPDD--YEAR (century omitted)/ Period/Day-of-Period

CCYYPPDD YEAR/Period/Day-of-Period

YYWW----YEAR (century omitted)/Week Number

CCYYWW--YEAR/Week Number

Numeric FISCAL date definitions:

------YY YEAR (century omitted)

----CCYY YEAR

----YYQQ YEAR (century omitted)/Quarter --CCYYQQ YEAR/Quarter

----YYPP YEAR (century omitted)/Period

--CCYYPP YEAR/Period

--YYPPDD YEAR (century omitted)/ Period/Day-of-Period

----YYWW YEAR (century omitted)/Week Number

--CCYYWW YEAR/Week Number

where:

- CC—Century number
- YY—Year number
- QQ—Quarter number
- PP—Period number
- WW—Week number
- DD—Day-of-Period number

025

INVALID FISCAL DATE INPUTOUTPUT FORMAT EXPECTED IN OUT 1/2/3 DATE MASK'

Reason:

An output date mask must be placed in the OUT1, OUT2, or OUT3 DATE MASK.

Action:

Specify one of the permitted date masks.

026

BAD (FISCAL) TO-DATE-MASK. SEE MANUAL

Reason:

The TO-DATE-MASK requires a Fiscal Calendar Mask.

Action:

Specify one of the permitted Fiscal Calendar Date Masks.

027

BAD (FISCAL) IN-NUMERIC-PARM, BEYOND TABLE.

Reason:

Special Fiscal Calendar Function F404 requires an IN-NUMERIC-PARM containing a signed (+/-) numeric value that specifies the number of days, weeks, periods, quarter, or years to decrement or increment a Fiscal Calendar date (FROM-DATE). The value specified is beyond the limits of the Fiscal Calendar table.

Action:

Review the table definition and make changes as required.

028

CALENDAR FORMAT EXPECTED IN OUT 1/2/3 DATE MASK

Reason:

The date mask(s) specified in OUT 1/2/3 date mask must be NON 360-DAY format.

029-039

360-DAY FORMAT NOT-ALLOWED or EXPECTED

Reason:

There are eight date formats that are intended for use only with Functions 301 through 304. They may not be used elsewhere, and they must be used correctly with the 301-304 series. These error messages elaborate on the proper usage and non-usage of these special formats.

```
029 -360-DAY FORMAT EXPECTED  IN OUT 1/2/3 DATE MASK
030 -360-DAY FORMAT NOT ALLOWED IN FROM-DATE-MASK
031 -360-DAY FORMAT EXPECTED  IN FROM-DATE-MASK
032 -360-DAY FORMAT NOT ALLOWED IN TO-DATE-MASK
033 -360-DAY FORMAT EXPECTED  IN TO-DATE-MASK
034 -360-DAY FORMAT NOT ALLOWED IN OUT1-DATE-MASK
035 -360-DAY FORMAT EXPECTED  IN OUT1-DATE-MASK
036 -360-DAY FORMAT NOT ALLOWED IN OUT2-DATE-MASK
037 -360-DAY FORMAT EXPECTED  IN OUT2-DATE-MASK
038 -360-DAY FORMAT NOT ALLOWED IN OUT3-DATE-MASK
039 -360-DAY FORMAT EXPECTED  IN OUT3-DATE-MASK
```

040

MASK NOT ALLOWED UPON USE AS INPUT

Reason:

This mask may not be used to describe input date information.

041

MASK NOT ALLOWED UPON USE AS OUTPUT

Reason:

This mask may not be used to describe output date information

042

AGE: FROM-DATE MAY NOT BE GREATER THAN TO-DATE

Reason:

For this function, the From-Date (birth date) must be less than or equal to the TO-DATE (date on which the age is being calculated).

043

COMPARE: FROM/TO MASKS MUST BE VALID AND EQUAL

Reason:

Both the From Date mask and To Date mask must be specified and contain the same values.

044

CONCATENATE CENTURY FROM DATE MASK INVALID

Reason:

There are ten date-format masks that are intended for use with the century concatenate function. One of the following date masks must be placed in FROM-DATE-MASK area.

```
YYMMDD-- --YYMMDD YYDDD--- YYKKK---
---YYDDD ---YYKKK YY------ YYMM----
----MMYY ------YY
```

045

TRUNCATE CENTURY FROM-DATE-MASK INVALID

Reason:

There are nine date-format masks that are intended for use with the Century Truncate function. One of the following date masks must be placed in the FROM DATE MASK area:

```
CCYYMMDD* CCYYDDD-CCYYKKK- -CCYYDDD
-CCYYKKK CCYYMM----CCYYMM CCYY----
----CCYY
```

Note: For FROM-DATE-MASK CCYYMMDD, the OUT1-DATE-MASK must contain YYMMDD--(alphanumeric format output) or --YYMMDD (numeric format output).

Execution Time Errors

This section describes messages 051-054, 099.

051

DATE BEFORE JAN 01, 0001

Reason:

During execution, there was an attempt to reference a date that was earlier than January 01, 0001.

Action:

You must modify your request so that such an attempt does not occur.

052

DATE AFTER DEC 31, 9999

Reason:

During execution, there was an attempt to reference a date that was later than December 31, 9999.

Action:

You must modify your request so that such an attempt does not occur.

053

DATE BEFORE PERIOD-START

Reason:

The answer would have been earlier than the specified period start.

Action:

None necessarily required.

054

DATE AFTER PERIOD-END

Reason:

The answer would have been later than the specified period end.

Action:

None necessarily required.

099

RESTRICTED OPERATION

Reason:

During execution, there was an attempt to execute a restricted operation. This can occur when END-PNTS-DEF is B or N; and the FROM-DATE is equal to the TO-DATE. This also can occur when the END-PNTS-DEF is B or N; and the DIRECTION/DISPLACEMENT is equal to zero.

Action:

You must modify your request so that such an attempt does not occur. See the appendix “End Points” for more information.

Systems Problems

This section describes messages 900-999.

900-999

9XX SYSTEM PROBLEM

Reason:

During execution, an internal error condition was detected due to program corruption.

Action:

You must speak to your System Administrator to learn what changes have been made since the initial product installation.

Appendix B: End Points

This appendix lists the end points that are used by CA Calendar Routines.

This chapter includes the following sections:

- Overview
- DAYSBETWEEN
- DATE +/- N DAYS

Overview

In Calendar calculations, the FROM-DATE and the TO-DATE are known as End Points. The END-PNTS-DEF Business Parameter specifies which of the two, if either, should be included in Calendar calculations.

There are four possible values for END-PNTS-DEF:

- B BOTH
- N NEITHER
- P FROM-DATE only
- T TO-DATE only

Restricted Operations

CA Calendar Routines restricts certain operations because they would yield ambiguous results. For example, an attempt to calculate the number of days between January 18, 2007 and January 18, 2007; neither end point can included.

There are two sets of restricted operations:

- Operations that calculate the Days Between.
  - END-PNTS-DEF = B or N
  - FROM-DATE and TO-DATE are equal
- Operations that calculate the Date +/- N Days.
  - END-PNTS-DEF = B or N and
  - Direction/Displacement = 0

Aside from the restricted operations, the interpretation of End Points is fairly straightforward. Several examples are given in the section Days Between. They involve variations on Calendar Day versus Processing Day, and Days Between versus Date +/- N.

DAYS BETWEEN

Find the number of days from August 17, 2007 to August 21, 2007; where 18 and 19 are not processing days.

Note: The answer is the last entry shown.

END-POINTS DEFINITION = B

Calendar Days		
Date	**Qual**	**Count**
20070817	YES	+1
20070818	YES	+2
20070819	YES	+3
20070820	YES	+4
20070821	YES	+5

Processing Days		
Date	**Qual**	**Count**
20070817	YES	+1
20070818	NO	+1
20070819	NO	+1
20070820	YES	+2
20070821	YES	+3

END-POINTS DEFINITION = N

Calendar Days		
Date	**Qual**	**Count**
20070817	N/A	+0
20070818	YES	+1
20070819	YES	+2
20070820	YES	+3
20070821	N/A	+3

Processing Days		
Date	**Qual**	**Count**
20070817	N/A	+0
20070818	NO	+0
20070819	NO	+0
20070820	YES	+1
20070821	N/A	+1

END-POINTS DEFINITION = F

Calendar Days		
Date	**Qual**	**Count**
20070817	YES	+1
20070818	YES	+2
20070819	YES	+3
20070820	YES	+4
20070821	N/A	+4

Processing Days		
Date	**Qual**	**Count**
20070817	YES	+1
20070818	NO	+1

Processing Days		
Date	**Qual**	**Count**
20070819	NO	+1
20070820	YES	+2
20070821	N/A	+2

END-POINTS DEFINITION = T

Calendar Days		
Date	**Qual**	**Count**
20070817	YES	+1
20070818	N/A	+0
20070819	YES	+1
20070820	YES	+2
20070821	YES	+3

Processing Days		
Date	**Qual**	**Count**
20070817	N/A	+0
20070818	NO	+0
20070819	NO	+0
20070820	YES	+1
20070821	YES	+2

DATE +/- N DAYS

Find the date that is +4 days from October 12, 2007, where 13 and 14 are not processing days.

Note: The answer is the last entry shown.

END-POINTS DEFINITION = B

Calendar Days		
Date	**Qual**	**Count**
20071012	YES	+1
20071013	YES	+2
20071014	YES	+3
20071015	YES	+4

Processing Days		
Date	**Qual**	**Count**
20071012	YES	+1
20071013	NO	+1
20071014	NO	+1
20071015	YES	+2
20071016	YES	+3
20071017	YES	+4

END-POINTS DEFINITION = N

Calendar Days		
Date	**Qual**	**Count**
20071012	N/A	+0
20071013	YES	+1
20071014	YES	+2
20071015	YES	+3

Calendar Days		
Date	**Qual**	**Count**
20071016	YES	+4
20071017	N/A	+4

Processing Days		
Date	**Qual**	**Count**
20071012	N/A	+0
20071013	NO	+0
20071014	NO	+0
20071015	YES	+1
20071016	YES	+2
20071017	YES	+3
20071018	YES	+4
20071019	N/A	+4

END-POINTS DEFINITION = F

Calendar Days		
Date	**Qual**	**Count**
20071012	YES	+1
20071013	YES	+2
20071014	YES	+3
20071015	YES	+4
20071016	N/A	+4

Processing Days		
Date	**Qual**	**Count**
20071012	YES	+1
20071013	NO	+1
20071014	NO	+1
20071015	YES	+2
20071016	YES	+3
20071017	YES	+4
20071018	N/A	+4

END-POINTS DEFINITION = T

Calendar Days		
Date	**Qual**	**Count**
20071012	N/A	+0
20071013	YES	+1
20071014	YES	+2
20071015	YES	+3
20071016	YES	+4

Processing Days		
Date	**Qual**	**Count**
20071012	N/A	+0
20071013	NO	+0
20071014	NO	+0
20071015	YES	+1
20071016	YES	+2
20071017	YES	+3
20071018	YES	+4

Index

G

H

I

L

M

N

O

P

R

T

W

Y