CA Calendar Routines®

Technical Manual

r6

Contents

Chapter 1: Overview

Chapter 2: Installation

Appendix C: Restricted Operations

Appendix D: Mnemonics

Appendix E: IBM MVS PL/1 and Assembler Program CALLs to CA Calendar Routines

Index

Chapter 1: Overview

This chapter provides background information about CA Calendar Routines and describes the role of the CA Calendar Routines administrator.

- Issues with Calendar Dates- A brief discussion of calendar date issues.
- CA Solutions- A brief discussion of the history and origin of the CA Calendar Routines.
- Role of system administrator- A brief description of your role as the technical person responsible for the CA Calendar Routines.

Issues with Calendar Dates

Data processing systems have typically taken a casual attitude toward calendar dates and the manner in which those dates are processed. Many organizations store dates in an inconsistent fashion. For example, it is not unusual to see a file that contains dates stored in MMDDYY and YYMMDD formats.

Recently, some data processing shops have had problems related to date processing. In 2000, some shops had difficulties with programs and files that were designed during the 1990's and stored only the last digit of the year. Four years later, some shops had trouble with systems that did not recognize 2004 as a leap year.

CA Solutions

CA Calendar Routines was written to address calendar-related problems. This product offers several calendar calculation routines, as well as features that can support the date calculation requirements of most data processing systems. Most notably, however, the product can handle any date between January 01, 0001 and December 31, 9999. To do so, it uses a special set of leap year rules. The existing leap year rules (two of which were established in 1582 by Pope Gregory XIII), generate three days of error every 10,000 years. Since CA committed to supporting all years with four digits, it had to define a new leap year rule to correct for the cumulative error. The resulting set of conversion processes is completely accurate for all dates within the range January 01, 0001 through December 31, 9999.

CA Calendar Routines is written in COBOL; and you have been provided with the source code. CA Calendar Routine uses conservative ANSI COBOL elements to facilitate installation on a variety of COBOL platforms; and they compile successfully using dialects such as IBM OS/VS COBOL, IBM VS COBOL II, and SAA COBOL.

Role of System Administrator

Your role, as the system administrator, is to provide technical support for the applications programming staff. This support involves installing and maintaining CA Calendar Routines and helping others to use CA Calendar Routines by providing ongoing application support.

The installation and maintenance tasks include:

- Defining holiday tables
- Defining special calendar fiscal tables
- Defining the environment
- Defining the business parameters

Installation and Maintenance

The first area of support concerns the installation and maintenance of CA Calendar Routines. There are a few tasks related to this undertaking. Each task is described briefly in the following list and all tasks are described in greater detail in later chapters.

Note: There are references to Special Fiscal Calendar tables and functions throughout this guide. Special Fiscal Calendars are an optional CA Calendar Routine processing feature. Bypass all Special Fiscal Calendar related reference of instructions if you are not using Fiscal Calendars.

Defining Holiday Tables

These tables contain the holidays that your company may want to consider. CA Calendar Routines provides several tables because, for example, you may want to dedicate one table to federal holidays and another table to state holidays. Also, the tables can be used to store special dates such as period start and end dates. Furthermore, it is possible for different applications within your company to have different work schedules and holiday requirements. Assigning unique holiday tables helps serve the needs of multiple applications.

Defining Special Fiscal Calendar Tables (Optional)

These tables contain the Fiscal Calendar definition that your organization has selected to use to support a special Fiscal Calendar, which does not quite coincide with the familiar twelve-month calendar (for which CA Calendar Routines is designed). Special Fiscal Calendars try to equalize the calendar into four-week or five-week periods (months) that begin and end at the start-of and end-of a week.

The CA Calendar Routines driver programs (TRCENGIN-batch and TRCENGNC-CICS) are shipped without Fiscal Calendar tables included or Fiscal Calendar functions enabled. You must install alternate versions of these programs (TRCENGFN-batch and TRCENGFC-CICS).

Note: The product is shipped with special Fiscal Calendar processing disabled. Installations using Fiscal Calendars can set the Fiscal Calendar version of the product as the default by:

- Renaming (or moving) programs TRCENGIN and TRCENGNC,
- Renaming program TRCENGFN to TRCENGIN and program TRCENGFC to TRCENGNC.

Defining the Environment

This involves general batch/online considerations as well as special requirements for accessing the holiday tables and the system current date and time.

Defining the Business Parameters

The business parameters are global settings that guide CA Calendar Routines through its calculations. As the system administrator, you are asked to assign defaults for the company and optionally various application groups.

Helping Others Use CA Calendar Routines

The second area of support concerns helping the applications staff use CA Calendar Routines. The best way to prepare yourself is to read the *Applications Manual*. By familiarizing yourself with the material, you will be able to answer their questions more effectively.

Chapter 2: Installation

This chapter describes requirements and procedures for installing CA Calendar Routines.

- Installation—Steps required to create, load, and compile the CA Calendar Routines programs.
- Defining Holiday tables—How you define holidays so that the invoking applications programs can reference them.
- Defining Special Fiscal Calendar tables—How you define the Fiscal Calendar so that the invoking application programs can reference Retail/ Manufacturing Calendars.
- Defining the environment—The hardware and software factors to consider for proper operation of CA Calendar Routines. This includes accessing the holiday tables and retrieving the system current-date and time.
- Defining business parameters—The global settings that guide CA Calendar Routines through its calculations.

How to Install CA Calendar Routines

To install CA Calendar Routines, perform the following steps:

1. Unload the product from the tape. See Step 2. Compile and Link the TRCENGIN, TRCENGNC (CICS), TRCSMALL, TRCFSGEN, and TRCHLGEN Programs for more information.
2. Modify the program TRCHLGEN to run at your site. See Step 2. Compile and Link the TRCENGIN, TRCENGNC (CICS), TRCSMALL, TRCFSGEN, and TRCHLGEN Programs for more information.
3. Compile and link TRCHLGEN

4. Run TRCHLGEN using the following JCL. To create a table you can

   – Create a dummy table by removing the three dates in the JCL

     Or

   – Change and add to the dates in the sample to create a real holiday table now. You can add more tables later.

```
//TSXXXXXX JOB (XXXX-XXX-XXX),'HOLIDAY TBL1 GENR',
//                      NOTIFY=XXXXXX,CLASS=A,MSGLEVEL=(1,0),MSGCLASS=X
/*JOBPARM    SYSAFF=*
//GENR001       EXEC PGM=TRCHLGEN
//STEPLIB        DD DSN=your. transcentury.loadlib,DISP=SHR
//HOLIDAYS    DD *
TBL-ID=01
20040101
20040704
20041225
/*
//PRINTOUT     DD SYSOUT=*
//COPYBOOK   DD  DSN=your.copybook..library(TRCHLT01), DISP=SHR
//HLPROGAM   DD DSN=your.source.library(TRCHLP01), DISP=SHR
//FLATINFO      DD DSN=flatfile.name, DISP=(NEW,KEEP),
DCB=(RECFM=FB,LRECL=8020,BLKSIZE=8020),
SPACE=(TRK,(1,1)),UNIT=SYSDA
//*
```

5. Modify TRCENGIN.

   You must scan through TRCENGIN and read the comments. The comments will direct you through customizing the program for your environment. Do a find on 19930517 to locate the lines. See Defining the Environment and Defining the Business Parameters for more information.

   Note: Be sure to comment out the date and time hard coded near the beginning of the program.

You are now ready to compile and link TRCENGIN.

Step 1. Create and Load COPYLIB and PROGRAM SOURCE Libraries

This section describes how to create and load the COPYLIB and PROGRAM SOURCE libraries.

Create and load the libraries

1. Copy the following JCL into your library and revise it to conform to your installation environment.

```
//UNLOAD JOB (accnt ###),'CA',MSGCLASS=X,MSGLEVEL=(1,1),
//  NOTIFY=userid,CLASS=A
//*******************************************************************/
//**  PRIOR EXECUTION MODIFY: ACCNT ###, USERID, TAPEVOL, USERNODE ****/
//**                    AND DISKVOL.                                ****/
//*******************************************************************/
//*
//*******************************************************************/
//*** INSTALL COPYLIB MEMBERS ***************************************/
//*******************************************************************/
```

```
//S01 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN1   DD DISP=OLD,DSN=TRC.TAPE6000.COPYLIB,
//  UNIT=TAPE,LABEL=(2,SL),VOL=SER=tapevol <= CHANGE
//OUT1   DD DSN=usernode.TCDS.COPYLIB,  <= CHANGE
//  DISP=(NEW,CATLG,DELETE),
//  SPACE=(TRK,(15,5,15)),UNIT=SYSDA, <= CHANGE
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//SYSIN  DD *
COPY OUTDD=OUT1,INDD=((IN1,R))
/*
//*******************************************************************/
//*** INSTALL PROGRAM SOURCE MEMBERS ********************************/
//*******************************************************************/
//S02 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN2   DD DISP=OLD,DSN=TRC.TAPE6000.SOURCE,
//  UNIT=TAPE,LABEL=(3,SL),VOL=SER=tapevol <= CHANGE
//OUT2   DD DSN=usernode.TCDS.SORCLIB,  <= CHANGE
//  DISP=(NEW,CATLG,DELETE),
//  SPACE=(TRK,(15,5,20)),UNIT=DISK, <= CHANGE
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//SYSIN  DD *
COPY OUTDD=OUT2,INDD=((IN2,R))
/*
```

usernode.TCDS.COPYLIB

- TRCCOMNP—This copybook contains the common Procedure Division routines that are used throughout the system. (Used by programs TRCHLGEN and TRCENGIN.)
- TRCCOMNW—This copybook contains the common Working Storage variables that are used throughout the system. (Used by programs TRCHLGEN and TRCENGIN.)
- TRCCONPD—This copybook contains the Procedure Division routines that are used whenever a CALL is made from the COBOL application program to the full function CA Calendar Routines (used by TRCENGIN or TRCENGNC).
- TRCCONVA—This copybook is the Conversational Area used by Assembler application programs to communicate with CA Calendar Routines.
- TRCCONVP—This copybook is the Conversational Area used by PL/1 application programs to communicate with CA Calendar Routines.
- TRCCONVR—This copybook is the Conversational Area that is used by COBOL application programs to communicate with CA Calendar Routines.
- TRCCONWS—This copybook contains the Working Storage variables used whenever a CALL is made from the COBOL application program to the full function Calendar Routines program (used by TRCENGIN or TRCENGNC).
- TRCERROR—This copybook contains the Procedure Division (supplied as an example) error routine that can be invoked should an error be detected during Calendar Routines processing.

- TRCFS444, TRCFS454—These copybooks contain sample Special Fiscal Calendar table definitions for the 4-4-4 and 4-5-4 fiscal calendars.
- TRCFSWSN, TRCFSWTN, TRCFSPN1, TRCFSPN2, TRCFSPN3, TRCFSPN4, TRCFSPN5—This series of copybooks is used by the CA Calendar Routines batch (TRCENGIN) and CICS (TRCENGNC) driver programs that *do not* contain Special Fiscal Calendar functions.

  Note: These are the default copybooks.
- TRCFSWTB, TRCFSPD1, TRCFSPD2, TRCFSPD3, TRCFSPD4, TRCFSPD5—This series of copybooks is used by the CA Calendar Routines batch (TRCENGFN) and CICS (TRCENGFC) driver programs that contain Special Fiscal Calendar functions.

  Note: These copybooks are only used in programs TRCENGFN and TRCENGFC.
- TRCFUNAM—This copybook contains a COBOL language table of CA Calendar Routines function numbers and names.
- TRCHLT01—This copybook contains the Working Storage variables used to define a holiday table (supplied as an example).
- TRCSMLPD—This copybook contains Procedure Division statements that can be included into an application to perform functions:
  - C101 (Days Between)
  - C102 (Date +/- N Days)
  - C201 (Get Calendar Next Day)
  - C202 (Get Calendar Previous Day)
  - U001 (Get Current Date/Time)
  - U002 (Validate Date)
  - U003 (Determine Leap Year)
  - U006 (Get Day of Week)
  - U007 (convert Date to Absolute Days)
  - U008 (convert Absolute Days to Date)
  - U009 (Reformat Date)
  - U011 (Compare Two Dates Same Format)
  - U012 (Concatenate Century)
  - U013 (Truncate Century)
- TRCSMLWS—This copybook contains the Working Storage variables used when performing functions.

- TRCU1XPD—This copybook contains Procedure Division statements that can be included into an application to perform the following functions:
  - U011 (Quick Compare)
  - U012 (Concatenate Century)
  - U013 (Truncate Century)—COBOL only
- TRCU1XWS—This copybook contains the Working Storage variables used when performing functions U011, U012, U013.

usernode.TCDS.SOURCE

- DATEMASK—This file contains the CA Calendar Routines date format masks used to create the Date Logic Generator ISPTLIB table DATEMASK.
- RESERVE—This file contains the COBOL reserved words used to create the Date Logic Generator ISPTLIB table 'RESERVE'.
- TRCENGFC—This program is the CICS engine version of CA Calendar Routines that includes access to Special Fiscal Calendars.
- TRCENGFN—This program is the engine of CA Calendar Routines that includes access to Special Fiscal Calendars.
- TRCENGIN—This program is the engine of CA Calendar Routines.
- TRCENGNC—This program is the CICS engine version of CA Calendar Routines.
- TRCHLGEN—This program is used to generate Holiday tables.
- TRCHLP98—This is a sample COBOL subprogram called by TRCENGIN, should your installation choose to access holiday tables from a subroutine.
- TRCHLRCV—This is a sample COBOL subprogram called by TRCENGIN, should your installation choose to store holiday tables on a VSAM file and use a subroutine to perform VSAM holiday file access.
- TRCHLRVS—This is a CICS COBOL module linked to by TRCENGNC. The CICS COBOL module reads holiday table information from a VSAM file.
- TRCSMALL—This program is the limited function engine version of CA Calendar Routines.

2. Run the job.

Step 2. Compile and Link the TRCENGIN, TRCENGNC (CICS), TRCSMALL, TRCFSGEN, and TRCHLGEN Programs

The PROGRAM SOURCE for all programs—located in the program source library shipped on the installation tape—conforms to COBOL 2 application programming standards. Modifications to these programs may be required for your specific installation. The COPYLIB members loaded in Step 1, Create and Load COPYLIB and PROGRAM SOURCE Libraries, will need to be accessible by your installation compiler that is used to compile the product components.

Two components will require compilation as one of the first steps of the installation process. Program TRCHLGEN generates the holiday table entries. Program TRCFSGEN generates the optional Fiscal Calendar table entries.

Important! *Do not* compile program TRCFSGEN if you are not going to use a special Fiscal Calendar.

Compile and link the programs

1. Compile and link programs TRCHLGEN and TRCFSGEN into a library that can be accessed by batch jobs and the TSO/REXX. (See Defining the Holiday Table and Defining the Special Fiscal Calendar Tables for more information.)
2. Compile and link batch processing programs TRCENGIN and TRCSMALL into a library that can be accessed by batch jobs and TSO ISPF (see the *CA Date Logic Generator Installation and Reference Manual* for more information).
3. If your installation has CICS, compile and link program TRCENGNC (CICS only) into a library that can be accessed by CICS.

   Note: The product is shipped with special Fiscal Calendar processing disabled. Installations using Fiscal Calendars can set the Fiscal Calendar version of the product as the default by:

   - Renaming (or moving) programs TRCENGIN and TRCENGNC
   - Renaming program TRCENGFN to TRCENGIN and program TRCENGFC to TRCENGNC.

Defining the Holiday Tables

CA Calendar Routines allows invoking application programs to reference up to 1296 different holiday tables. As the System Administrator, you must define all tables that the user may request (plus 01-10 as discussed in the following note). Some possible uses include federal holidays, state holidays, bank holidays, and stock market holidays (domestic and/or foreign). Also, holiday tables can be used to store special dates such as period start and end dates.

Note: If you use an earlier release of CA Calendar Routines, the number of tables and their IDs were fixed at 01-10 and were mandatory even if some or all were empty. Since some of your current users may be referring to those holiday tables, you must create all of them as you had been, so they will have upward compatibility. If you do not do this, they may begin receiving error code 001 (BAD HOLIDAY-TBL. COULD NOT BE ACCESSED) and/or receiving different calculations from CA Calendar Routines than they had been.

Components Used to Define a Holiday Table

Each major component used to define a holiday table is described in the following list.

TRCHLGEN

TRCHLGEN is the program that processes all of the input holidays. It is provided as program source, and is ready to be executed once you have compiled and linked the program.

Note: You may have to change minor items such as using a quote rather than an apostrophe, GOBACK, STOP RUN, or EXIT PROGRAM, or INCLUDE rather than COPY.

TRCHLGEN reads raw holiday information and refines it for efficient access during execution. The final result is a copybook, a single flatinfo-record and a holiday-program containing the copybook. The copybook can be used to access the holidays internally through working storage. The flatinfo-record can be used to access the holidays externally once it is loaded by a utility to a file, table, or database. The holiday-program containing the copybook can be called dynamically from the CA Calendar Routines engine.

Use of the copybook, flatinfo-record, and holiday-program are not mutually exclusive. Your shop may want to store the most frequently used tables as copybooks, and allow the less frequently used tables be updated as an external table.

TRCENGIN and TRCENGNC (optional-CICS only)

TRCENGIN and the CICS version of TRCENGNC are the main programs in CA Calendar Routines. They can access the holiday information from either of these sources:

- From working storage using the copybook
- Externally from a file, table, or database
- From the holiday program that can be called dynamically

HOLIDAY File

The HOLIDAY File is the input to the program. The first record in the input file contains the table identification in the format TBL-ID=xx, where xx must be alphanumeric, is required and will identify the table number that you are creating. The rest of the records contain one holiday in columns 1-8 and must be a valid date in CCYYMMDD format. Comment cards (an asterisk in position one of the record) can be inserted at any location in the Holiday File.

Note: You may only create one table at a time.

PRINT File

The PRINT File provides you with hardcopy results. The ERROR REPORT will inform you of edit errors and duplicate holidays. The CONTROL REPORT will tell you the final status and statistics of the program. The CALENDAR LISTING will be provided if there was input with no errors. Information printed will range from January 1 of the earliest input year to December 31 of the latest input year; the report will be marked with the input holidays.

COPYBOOK File

The COPYBOOK File contains detail holiday information related to the holidays that you specified. The holidays are stored in ascending sequence according to the absolute days format (for example, the date 99991231 is 3652056 days from Jan 1, 0001). This provides maximum efficiency during the execution of program TRCENGIN. The name of the copybook that you create should be TRCHLT xx, where xx is the identifier coded on the TBL-ID=xx card.

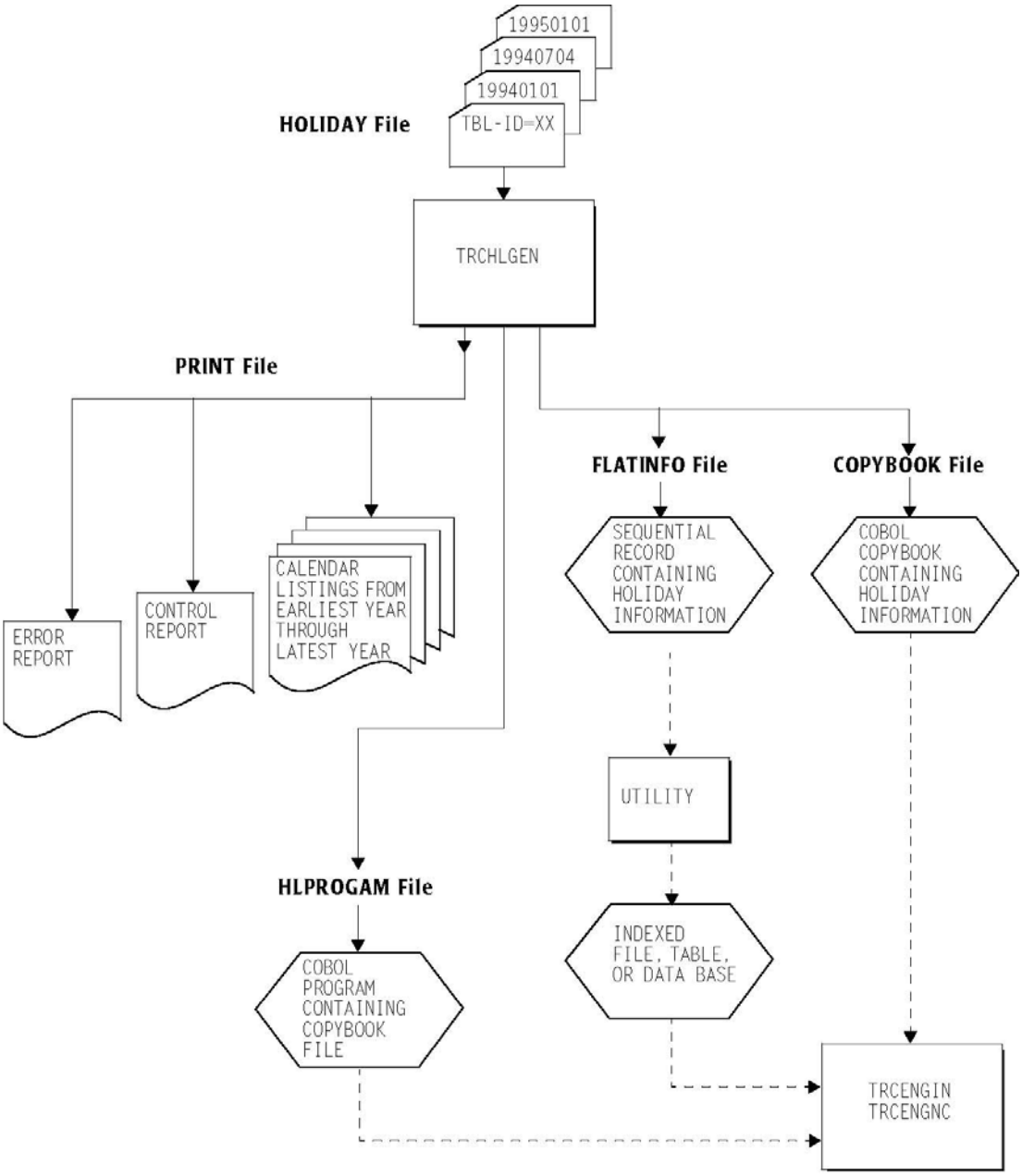
FLATINFO File

The FLATINFO File and the COPYBOOK File contain identical information. Each record contains a single holiday date that is then loaded into an object such as an indexed file, a table, or a database and accessed as an external source when the engine program needs to reference holiday information.

HLPROGAM File

The HLPROGAM File contains a program that contains the same copybook as the COPYBOOK File. The HLPROGAM program wraps the copybook around a program so that it can be called dynamically by the CA Engine to pass holiday information back during execution. It has the advantage of quick access during execution, and flexibility of maintaining holiday information without disturbing the CA Engine (for example, recompiling with the new copybook or having to maintain a file FLATINFO FILE).

The following illustration shows how the previous components fit into the process of defining a holiday table.



Creating a Table

You can create an empty holiday table or one that contains holidays. Procedures for creating either type of table are described in Generating an Empty Holiday Table and Generating a Holiday Table with Holidays.

Generating an Empty Holiday Table

You may need to create empty tables when running an older version of this product. To create an empty table, execute the program TRCHLGEN with only the TBL-ID=xx record as input.

The Holiday Table Generation Options table shows you the results of not entering any holiday dates. You will not receive an error report because no holiday input exists. Next, you receive a control report with MSG-004. Finally, you receive a copy/prog/flatinfo-record, representing an empty holiday table.

Generating a Holiday Table with Holidays

If you want to create a table that contains holidays, use the following procedure.

To create a table containing holidays

1. Decide the earliest year and latest year for which you want to create holidays.
2. Compile, link, and execute TRCHLGEN, providing two dates: January 1 of the earliest year and January 1 of the latest year. The program will generate calendar listings that range from January 1 of the earliest year through December 31 of the latest year. You can use these listings to identify and mark all of the holidays that need to be generated.

3. Compile, link, and execute TRCHLGEN, providing all of the calendar dates that you want to identify as holidays. If the program does not execute successfully at this point, you might need to modify the JCL or add a RECORDING MODE F clause to each FD statement for all files defined in the TRCHLGEN program. Otherwise, the program will execute successfully, in which case there are three possibilities.

   - If the TBL-ID was not alphanumeric, you will not receive a calendar listing. You will receive a control report (with MSG-000), stating that the TBL-ID is in error and must be alphanumeric.
   - If the input was valid, you will not receive an error report, since no errors exist. You will receive a control report (with MSG-005) and a calendar listing ranging from January 1 of the earliest year through December 31 of the latest year with all of your holidays marked. You will also receive a copy/prog/flatinfo-record that contains all of the holiday information required by program TRCENGIN.
   - If the input was bad, you will not receive a calendar listing or a copy/prog/flatinfo-record. You will receive a control report (with MSG-006) and a detailed error report showing you all of the errors that were detected.

Holiday Table Generation Options

Generation Options		Holiday File		
		Input Is Provided		
		No Input	Good Input	Bad Input
Print File	Error Report	No	No	Yes
	Control Report	Yes	Yes	Yes
	Message Number	004	005	006
	Calendar Listings	No	Yes	No
Copy/Prog/FlatInto		Yes	Yes	No

Defining the Special Fiscal Calendar Tables

CA Calendar Routines allows invoking applications to reference a four-week, thirteen period calendar (4-4-4) and a twelve period calendar with quarters containing a four week period, five week period, and a four week period (4-5-4). The 4-4-4 and 4-5-4 calendar tables shipped with this version of the product are referenced by CA Calendar Routines.

Components Used to Define a Special Fiscal Calendar Table

This section describes each major component used to define a fiscal calendar table. The Fiscal Calendar tables for 2007 show how these components fit into the process of defining a fiscal calendar table.

TRCFSGEN

TRCFSGEN is the program that processes all input fiscal calendar table input. The output from this program is:

- a copybook
- external flat file
- a fiscal-program which can be called dynamically

FISCAL CALENDAR FILE

The Fiscal Calendar File is the input to the program that contains calendar entries defining each fiscal year. The fiscal calendar Input includes entries defining four fiscal calendar tables:

- Fiscal Year Ending
- Fiscal Quarter Ending
- Fiscal Period Ending
- Fiscal Week Ending

Input to each of these tables contains the Fiscal Calendar Year number (format CCYY), a table type designator code, a table period number, and the standard Gregorian date (format CCYYMMDD) of the ending date.

The following table definition records are used to define Fiscal Calendar tables for 2007:

```
         1        2
----5----0----5----0 (record positions)
20070    19990102 <======== Fiscal Year (type '0' - position 5,
                                          period number blank(2)
                                          for year - positions 6-7,
                                          Gregorian date 8-15)
200710120070328 <======== Fiscal Quarters
200710220070620                    (type '1', quarters '01-04',
200710320070912                     and Gregorian dates)
200710419990102
200720120070131 <======== Fiscal Periods
200720220070228                   (type '2', periods '01-13',
200720320070328                    and Gregorian dates)
200720420070425
200720520070523
200720620070620
200720720070718
200720820070815
200720920070912
200721020071010
200721120071107
200721220071205
200721319990102
----5----0----5----0 (record positions)
200730120070110 <======== Fiscal Weeks
200730220070117                          (type '3', periods '01-52',
200730320070124                            and Gregorian dates)
200730420070131
200730520070207
200730620070214
200730720070221
200730820070228
200730920070307
200731020070314
200731120070321
200731220070328
200731320070404
200731420070411
200731520070418
200731620070425
200731720070502
200731820070509
200732520070627
200732620070704
200732720070711
200732820070718
200732920070725
200733020070801
200733120070808
200733220070815
200733320070822
200733420070829
200733520070905
200733620070912
200733720070919
2007338
20070926
2007339
20071003
20073501
0071219
2007351
20071226
2007352
19990102
```

TRCENGIN

TRCENGIN is the main program in CA Calendar Routines. TRCENGIN is accessed during the fiscal table creation process when obtaining:

- Absolute day values
- Beginning/ending dates
- Various date output formats

COPYBOOK File

The COPYBOOK File contains detailed fiscal calendar information related to the definitions of fiscal year, quarter, period, and week-end dates.

The name of the copybook that you create should be TRCFS444, where 444 is the identifier coded for the 4-4-4 Fiscal Calendar table. Revise the COPY TRCFSNUL" TRCENGIN (and/or TRCENDNC) COBOL source statement to COPY TRCFS444.

FLATINFO File

The FLATINFO File and COPYBOOK File contain identical information. Each record contains a single fiscal calendar table date. The entries in this file are loaded into an object such as an indexed file, a table, or a database and accessed as an external source when the engine program needs to reference fiscal calendar information.

FSPROGAM File

The FSPROGAM File contains program source code that references the same copybook as the COPYBOOK File. The FSPROGAM program wraps the copybook around a program so that it can be called dynamically by the CA Calendar Routines engine to pass fiscal calendar table information back during execution.

This subprogram has the advantage of quick access during execution and flexibility of maintaining Fiscal Calendar information without disturbing the CA Calendar Routines engine (that is, recompiling with the new copybook or having to maintain a file FLATINFO File).

REXX EXEC execution

A REXX EXEC is shipped on the Date Logic Generator installation media that is used to generate input records to create the Fiscal 4-4-4 or 4-5-4 Calendar tables. This REXX EXEC member (FSCL444 or FSCL454) can be executed from the ISPF Command Shell by entering the highlighted text and pressing Enter:

```
Menu    List    Mode    Functions    Utilities    Help
                                      ISPF Command Shell
Enter TSO or Workstation commands below:
===> ex 'usernode.exec(FSCl444) $444 2050 2052 usernode.pds.data(ip444)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
usernode.pds.copylib(TRCFS444)'
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

A text message is displayed requesting which portion of the REXX EXEC you want to start executing.

Enter C and press Enter. The REXX EXEC will create two data files:

- Data file used to create the Fiscal 4-4-4 table.

```
File Edit Confirm Menu Utilities Compilers Test Help
VIEW   usernode.PDS.data(TRCFS444) - 01.00 Columns 00001 00072
Command ===>   Scroll ===> CSR
000001 TBL-ID=$444
000002 20500 20501231
000003 205010120500326
000004 205010220500618
000005 205010320500910
000006 205010420501231
000007 205020120500129
000008 205020220500226
```

- COPYLIB file used by TRCENGIN to access the Fiscal 4-4-4 table.

```
VIEW USERNODE.PDS.COPYLIB(TRCFS444) - 01.00        Columns 00001 00072
Command ===>   Scroll ===> CSR
****** ***************************** Top of Data
******************************
000001 /
000002 *
000003   *-------------------------------------------------------------*
000004 *
000005 * THIS COPYBOOK CONTAINS FISCAL PERIOD END DATES IN ABSOLUTE
000006 * DAYS FORMAT. THE DATES REPRESENT YEAR, QUARTER, PERIOD AND
000007 * WEEK ENDINGS. THE TYPE OF PERIOD, GREGORIAN, AND
000008 * ABSOLUTE DATE ARE ALSO LISTED.
000009 *
000010 *
000011 * TYPE   GREGORIAN ABSOLUTE
000012 *
000013 * 2050 YEAR    20501231 00748747
000014 * 2051 YEAR    20511230 00749111
000015 * 2052 YEAR    20521228 00749475
000016 *
000017 * 2050 QUARTER 01 20500326 00748467
000018 * 2050 QUARTER 02 20500618 00748551
000019 * 2050 QUARTER 03 20500910 00748635
000020 * 2050 QUARTER 04 20501231 00748747
000021 * 2051 QUARTER 01 20510325 00748831
000022 * 2051 QUARTER 02 20510617 00748915
000023 * 2051 QUARTER 03 20510909 00748999
000024 * 2051 QUARTER 04 20511230 00749111
000025 * 2052 QUARTER 01 20520323 00749195
000026 * 2052 QUARTER 02 20520615 00749279
000027 * 2052 QUARTER 03 20520907 00749363
000028 * 2052 QUARTER 04 20521228 00749475
000029 *
000030 * 2050 PERIOD 01 20500129 00748411
000031 * 2050 PERIOD 02 20500226 00748439
- - - - - - - - - Omitted Display Lines - - - - - - - - - - - - - - - - - -
000230   *-------------------------------------------------------------*
000231 *
000232 *
000233    01 TRC-FSCL444-DTL-TABLE.
000234 05 TRC-FSCL444-YR-TABLE.
000235 10 TRC-FSCL444-YR-COUNT PIC S9(8) COMP VALUE +00000005.
000236 10 TRC-FSCL444-YR-DEF.
000237    15 FILLER PIC 9(04) VALUE  0000.
000238    15 FILLER PIC S9(08) COMP VALUE -99999999.
000239    15 FILLER PIC 9(04) VALUE  2050.
000240    15 FILLER PIC S9(08) COMP VALUE +00748747.
000241    15 FILLER PIC 9(04) VALUE  2051.
000242    15 FILLER PIC S9(08) COMP VALUE +00749111.
000243    15 FILLER PIC 9(04) VALUE  2052.
000244    15 FILLER PIC S9(08) COMP VALUE +00749475.
000245    15 FILLER PIC 9(04) VALUE  9999.
000246    15 FILLER PIC S9(08) COMP VALUE +99999999.
000247 05 TRC-FSCL444-QT-TABLE.
000248 10 TRC-FSCL444-QT-COUNT PIC S9(8) COMP VALUE +0000001
000249 10 TRC-FSCL444-QT-DEF.
000250    15 FILLER PIC 9(04) VALUE  0000.
000251    15 FILLER PIC 9(02) VALUE  00.
000252    15 FILLER PIC S9(08) COMP VALUE -99999999.
000253    15 FILLER PIC 9(04) VALUE  2050.
000254    15 FILLER PIC 9(02) VALUE  01.
000255    15 FILLER PIC S9(08) COMP VALUE +00748467.
```

Batch Job Execution

This section describes the batch method of creating viable Fiscal Calendar entries.

To create fiscal calendar entries using a batch job

1. Determine the earliest year and latest year for which you want Fiscal Calendar entries.
2. Subtract one from the earliest year and add one to the latest year.
3. Enter ending dates that represent the Fiscal Calendar Year ending date, quarter ending dates (4), period ending dates (13), and week ending dates (52 or 53) for each year of the Fiscal Calendar in the format described in Fiscal Calendar Input File.
4. Execute TRCFSGEN using the input file from Step 1. If the TRCFSGEN program does not execute successfully, you may need to modify the JCL or the TRCFSGEN program to make it specific to your COBOL program environment.

   The following is a sample TRCFSGEN Execution JCL:

```
//jobname JOB (acct),'TRCFSGEN',TIME=1440,
// CLASS=A,MSGCLASS=X,NOTIFY=userid
//*
//*************************************************
//*
//* TEST TRCFSGEN FISCAL TABLE GENERATION
//*************************************************
//*
//STEP1 EXEC PGM=TRCFSGEN,REGION=4M
//STEPLIB DD DSN=SYS1.COB2LIB,DISP=SHR
// DD DSN=SYS1.COB2COMP,DISP=SHR
// DD DSN=usernode.LOADLIB,DISP=SHR
//SYSMSG DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSLIST DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSDBOUT DD SYSOUT=*
//FISCAL DD DSN=usernode.input.file,DISP=SHR
//COPYBOOK DD DSN=usernode.COPYBOOK.OUTPUT,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6240)
// UNIT=SYSDA,
// SPACE=(TRK,(1,1),RLSE),
// DISP=(,CATLG)
//FSPROGAM DD DSN=usernode.PROGRAM.OUTPUT,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6240)
// UNIT=SYSDA,
// SPACE=(TRK,(1,1),RLSE),
// DISP=(,CATLG)
//FLATINFO DD DSN=usernode.FLATINFO.OUTPUT,
// DCB=(RECFM=FB,LRECL=8020,BLKSIZE=8020)
// UNIT=SYSDA,
// SPACE=(TRK,(1,1),RLSE),
// DISP=(,CATLG)
```

Upon successful execution, three files are created:

- COPYBOOK File—See COPYBOOK FILE for more information.
- FLATINFO File—See FLATINFO FILE for more information.
- FSPROGAM file—See FSPROGAM FILE for more information.

Defining the Environment

The following sections describe how to define aspects of the CA Calendar Routines environment.

Global Changes and Invocation Protocol

As you install CA Calendar Routines you will need to consider the environment in which CA Calendar Routines is used. You may have the same general considerations in TRCENGIN as you did in TRCHLGEN (for example, quote instead of apostrophe, GOBACK or STOP RUN instead of. EXIT PROGRAM, or COPY instead of INCLUDE). Additionally, your shop may have particular coding standards. Next, you will have to consider the protocol between the invoking programs and CA Calendar Routines. The following table is provided as an example of how to adapt CA Calendar Routines to a batch environment (program TRCENGIN or TRCSMALL) and to a CICS environment (program TRCENGNC).

Note: In the batch environment, you must specify GOBACK rather than EXIT PROGRAM in TRCENGIN and TRCSMALL (this is the default). In the CICS environment, LINK is recommended (over CALL) so that the linked program can execute CICS commands.

Batch	**CICS**
Application Program	**Application Program**
WORKING-STORAGE SECTION. COPY TRCCONVR.	WORKING-STORAGE SECTION. COPY TRCCONVR.
PROCEDURE DIVISION.	PROCEDURE DIVISION.
CALL "TRCENGIN" or “TRCSMALL” USING TRC-CONVR-CONVERSATIONAL.	EXEC CICS LINK PROGRAM("TRCENGNC") COMMAREA(TRC-CONVR-CONVERSATIONAL) LENGTH(1000) END-EXEC.
GOBACK.	EXEC CICS RETURN END-EXEC.
TRCENGIN or TRCSMALL	**TRCENGNC**

Batch	CICS
WORKING-STORAGE SECTION. COPY TRCCONVR.	WORKING-STORAGE SECTION. COPY TRCCONVR.
LINKAGE SECTION. 01 TRC-CONVERSATIONAL-LINKAGE PIC X(1000)	LINKAGE SECTION. 01 DFHCOMMAREA. 05 TRC-CONVERSATIONAL-LINKAGE PIC X(1000)
PROCEDURE DIVISION USING TRC-CONVERSATIONAL-LINKAGE	PROCEDURE DIVISION.
MOVE TRC-CONVERSATIONAL-LINKAGE TO TRC-CONVR-CONVERSATIONAL.	MOVE TRC-CONVERSATIONAL-LINKAGE TO TRC-CONVR-CONVERSATIONAL.
GOBACK.	EXEC CICS RETURN END-EXEC.

Null Character Date Mask Validation

The Date Mask Validation indicator can be set for all applications in your organization by changing the value of a data element in program working storage. This element must be changed in the programs TRCENGIN, TRCSMALL, and TRCENGNC (if you use the CICS version of TRCENGIN). The usual setting for data element TRC-ENHNC-EDIT-NULL-SW is N.

The null character (-) nominally means test for leading zeros or trailing blanks. If you have special needs to process unconverted, mixed data, or a combination of both data types, set the value of the following data element to N (as shown) and refer to the CA Calendar Routines Business parameters described in the *Application Manual*.

```
01 TRC-ENHNC-EDIT-NULL-SW  PIC X(01) VALUE 'N'.
88 TRC-ENHNC-DO-NOT-EDIT-NULLS VALUE 'N'.
88 TRC-ENHNC-EDIT-NULLS VALUE 'Y'.
```

However, due to an error in versions 3.1 and 4.0, the following date formats have not been complied with the edit nulls criteria. This error has been corrected, but users upgrading to this release from 3.1 or 4.0 may notice the symptoms described below.

```
--YYMMDD YYMMDD—
--MMDDYY MMDDYY—
--DDMMYY DDMMYY--
```

It is likely that you have been populating the input variables with COBOL numeric or alphanumeric MOVE statements. If you have been using the MOVE statements in this fashion, then you have been right-justify zero-filling and left-justify space-filling respectively anyway, in which case you will observe no effect from this corrected bug.

However, if you have been populating the fields using other methods, and these methods did not contain leading zeros or trailing spaces as required, then you will begin to receive an error message that the input variable does not conform to editing specifications.

To correct this validation inconsistency, ensure that the value of the data element is set to Y (as shown):

```
01 TRC-ENHNC-EDIT-NULL-SW PIC X(01) VALUE 'Y'.
88 TRC-ENHNC-DO-NOT-EDIT-NULLS VALUE 'N'.
88 TRC-ENHNC-EDIT-NULLS VALUE 'Y'.
```

Set Alphabetic Output Case (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 CA Calendar Routines to output all alphabetic literals in uppercase (such as JANUARY 1, 2007). Change this value to M if you want CA Calendar Routines to output date literals in mixed case (such as Monday January 8, 2007).

TRCENGIN/TRCSMALL Dynamic Calls - Static Calls

The TRCENGIN and TRCSMALL can be called dynamically or statically.

Dynamic Call Environment

Batch application programs that invoke the TRCENGIN or TRCSMALL program can ensure execution of the latest version of the program by calling these programs dynamically. Dynamic calls force the application program using TRCENGIN or TRCSMALL to load a fresh copy of the program into computer memory when the initial call is made to the Calendar Routines program.

Syntax

```
01 LINK-TO-PROGRAM PIC X(08) VALUE "TRCENGIN".
CALL LINK-TO-PROGRAM  USING TRC-CONVR-CONVERSATIONAL.
```

Static Call Environment

Programs that are statically called load any called programs into the executable module during module creation (such as COBOL batch program compile and link). Any changes made to the called program (such as TRCENGIN) after application programs have been compiled and linked will require recompilation and/or relinkage of all programs that use the TRCENGIN or TRCSMALL program.

Syntax

```
CALL "TRCENGIN" USING TRC-CONVR-CONVERSATIONAL.
CALL "TRCSMALL" USING TRC-CONVR-CONVERSATIONAL.
```

Guaranteeing Dynamic Call Invocation

As previously described, the TRCENGIN or TRCSMALL program can be called dynamically and statically. A method that will force all (batch) programs using the TRCENGIN or TRCSMALL program to call the program dynamically is shown next.

To specify dynamic call invocation

1. Rename the TRCENGIN and/or TRCSMALL programs to a secret name (such as TRCDYENG).
2. Create a stub program that replaces TRCENGIN:

```
ID DIVISION.
PROGRAM-ID. TRCENGIN.
....
....
WORKING-STORAGE SECTION.
....
....
01 PROGRAM-NAME  PIC X(08) VALUE "TRCDYENG".
....
....
LINKAGE SECTION.
* COPY TRCCONVR.
*
01 TRC-CONVERSATIONAL-AREA.
 05 TRC-....
.......
.......
PROCEDURE DIVISION USING TRC_CONVERSATIONAL-AREA.
CALL PROGRAM-NAME USING
TRC-CONVERSATIONAL-AREA.
EXIT PROGRAM.
```

Any application program calls made to the previous program (TRCENGIN) will always force dynamic invocation of the CA Calendar Routines processing program.

Current Date Retrieval

Another area that will require your attention is related to current date and time retrieval. TRCENGIN contains several examples of how to retrieve the current date from the system and several examples are shown next. In programs TRCENGIN, TRCSMALL, and TRENGNC, they can be found at CUSTOMIZATION PART 1 of working storage and the procedure division. In the CICS TRCENGNC program, you may retrieve current-date from EIBDATE, saving an EXEC CICS.

Example 1—Retrieve Current Date (VS COBOL 2)

```
WORKING-STORAGE SECTION.
01 TRC-ANSI-DATE    PIC 9(06).
01 TRC-ANSI-DATE-REDEF REDEFINES TRC-ANSI-DATE.
05 TRC-ANSI-YEAR     PIC 9(02).
05 TRC-ANSI-MONTH     PIC 9(02).
05 TRC-ANSI-DAY   PIC 9(02).
01 TRC-CURDT-LS-CCYYMMDD.
05 TRC-CURDT-LS-CC PIC 99.
05 TRC-CURDT-LS-YY PIC 99.
05 TRC-CURDT-LS-MM PIC 99.
05 TRC-CURDT-LS-DD PIC 99.
PROCEDURE DIVISION.
ACCEPT TRC-ANSI-DATE FROM DATE.
* SPECIAL REGISTER DOES NOT CONTAIN CENTURY.
* THEREFORE ASSUME INSTALLATION 1993 OR BEYOND.
IF TRC-ANSI-YEAR > 92
MOVE 19 TO TRC-CURDT-LS-CC
ELSE
MOVE 20 TO TRC-CURDT-LS-CC.
MOVE TRC-ANSI-YEAR TO TRC-CURDT-LS-YY.
MOVE TRC-ANSI-MONTH TO TRC-CURDT-LS-MM.
MOVE TRC-ANSI-DAY  TO TRC-CURDT-LS-DD.
EXIT PROGRAM.
```

Example 2—Retrieve Current Date (CICS)

```
WORKING-STORAGE SECTION.
01 TRC-ASKTIME-FORMATTIME-AREAS.
05 TRC-AF-ABSTIME   PIC S9(15) COMP-3.
05 TRC-AF-CENTURY-YEAR    PIC S9(08) COMP.
05 TRC-AF-MONTH PIC S9(08) COMP.
05 TRC-AF-DAY  PIC S9(08) COMP.
01 TRC-CURDT-RF-CCYYMMDD.
05 TRC-CURDT-RF-CCYY   PIC 9(04).
05 TRC-CURDT-RF-MM    PIC 9(02).
05 TRC-CURDT-RF-DD    PIC 9(02).
PROCEDURE DIVISION.
EXEC CICS ASKTIME ABSTIME (TRC-AF-ABSTIME) END-
EXEC. EXEC CICS FORMATTIME ABSTIME  (TRC-AF-ABSTIME)
YEAR   (TRC-AF-CENTURY-YEAR)
MONTHOFYEAR (TRC-AF-MONTH)
DAYOFMONTH (TRC-AF-DAY) END-EXEC.
 MOVE TRC-AF-CENTURY-YEAR TO TRC-CURDT-RF-CCYY.
MOVE TRC-AF-MONTH    TO TRC-CURDT-RF-MM.
MOVE TRC-AF-DAY  TO TRC-CURDT-RF-DD.
EXEC CICS RETURN END-EXEC.
```

Holiday Table Retrieval

Another area that will require your attention is related the method of retrieving holiday information. CA Calendar Routines enables you to access holiday information internally or externally. These areas can be found in the TRCENGIN program and the TRCENGNC program at CUSTOMIZATION PART 2 of working storage and the procedure division.

Accessing Holiday Information

If you choose to access holiday information using the internal method, you must reference the copybooks that were created from TRCHLGEN.

If you choose the external methods, you must either:

- Link to a program that contains the copybook with fiscal calendar information (FSPROGAM output from TRCFSGEN)
- Access the flatinfo-record that was originally created from TRCFSGEN, and has been subsequently loaded into a file, table, or database.

You may be able to access the information directly from TRCENGNC if the appropriate file is available to the environment (for example, CICS had been brought up, or the database had been started). Otherwise, you may find it necessary to invoke a *third* program to read the information and pass it back to TRCENGNC or TRCENGIN. A program of this type might also require logic and invocations to open and close the file.

Example 1—Internal Retrieval (COPYBOOKS)

```
WORKING-STORAGE SECTION.
COPY TRCHLT01.
COPY TRCHLT02.
PROCEDURE DIVISION.
IF TRC-INTRL-HOLIDAY-TBL-01
MOVE TRC-HLT01-DTL-TABLE
TO TRC-HLTBL-DTL-TABLE
ELSE
IF TRC-INTRL-HOLIDAY-TBL-02
```

Example 2—External Retrieval (CICS VSAM READ)

```
WORKING-STORAGE SECTION.
01 TRC-HLRTV-DTL-RECORD.
05 TRC-HLRTV-DTL-KEY.
05 TRC-HLRTV-DTL-TABLE.
PROCEDURE DIVISION.
EXEC CICS
READ DATASET ("FILENAME")
INTO (TRC-HLTRV-DTL-RECORD)
RIDFLD (TRC-HLTRV-DTL-KEY)
END-EXEC.
IF THE RESULT IS OK (HANDLE CONDITION, RESP ETC)
MOVE TRC-HLRTV-DTL-TABLE
TO TRC-HLTBL-DTL-TABLE
ELSE
ERROR PROCESSING.
```

Special Fiscal Calendar Table Retrieval

CA Calendar Routines enables you to access Special Fiscal Calendar information internally or externally. These areas can be found in the TRCENGIN (TRCENGFN/TRCENGFC) program at CUSTOMIZATION PARTS of Working Storage and the Procedure Division.

Accessing Fiscal Calendar Information

If you choose to access fiscal table information using the internal method, you must reference the copybooks that were created from TRCFSGEN and/or REXX EXEC FSCL444 or FSCL454.

If you choose the external methods, you must either:

- Link to a program that contains the copybook with fiscal calendar information (FSPROGAM output from TRCFSGEN)
- Access the flatinfo-record that was originally created from TRCFSGEN, and has been subsequently loaded into a file, table, or database.

You may be able to access the information directly from TRCENGIN if the appropriate file is available to the environment (for example, CICS has been brought up, or the database has been started). Otherwise, you may find it necessary to invoke a third program to read the information and pass it back to TRCENGIN. A program of this type might also require logic to open and close a file.

```
BROWSE  PDGBO.WRK6000.SOURCE(TRCENGIN) - 01.99  Line 00019007 Col 001 080
Command ===>   Scroll ===> CSR
WORKING-STORAGE SECTION.
COPY TRCFS444 SUPPRESS.
PROCEDURE DIVISION.
*
930000-RETRIEVE-FISCAL-INFO.
*
****************************************************************
*
* USE THIS CODE IF FISCAL YEAR TABLES ARE IN WORKING STORAGE
*
****************************************************************
*
IF TRC-CONVR-FISCAL-ID = '$444'
MOVE TRC-FSCL444-YR-TABLE
TO TRC-FISCAL-YEAR-TABLE
MOVE TRC-FSCL444-QT-TABLE
TO TRC-FISCAL-QTR-TABLE
MOVE TRC-FSCL444-PR-TABLE
TO TRC-FISCAL-PER-TABLE
MOVE TRC-FSCL444-WK-TABLE
TO TRC-FISCAL-WK-TABLE
```

Defining the Business Parameters

CA Calendar Routines performs a variety of functions. Most of these functions are controlled by a set of twelve parameters known as business parameters.

How the System Retrieves Parameter Values

There are two ways that the CA engine can get the values for the business parameters:

- The programmer can use the default values pre-assigned in the communications area
- The programmer can assign values explicitly

Assigning Parameter Values

As the System Administrator, you must assist with the first choice by assigning the initial values of the business parameters in the communications area, TRCCONVR (see the following example), TRCCONVP (PL/1), or TRCCONVA (Assembler). You may find it helpful to define one or more copies of TRCCONVR so that you can accommodate major applications individually. For example, suppose the Accounting Department has a July Fiscal Year and works Monday through Friday on a Calendar Year basis, but the Shipping Department works Monday through Saturday. In this case, it might be easier to provide each department with its own copy of TRCCONVR.

In some rare instances your shop may decide that the invoking application programs should not be allowed to control a particular Business Parameter. If this happens, you can convert the corresponding field in TRCCONVR to FILLER and assign the parameter a value unconditionally in the TRCENGIN program.

```
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 ' '.
10 FILLER    PIC X(18) VALUE SPACE. .. . . . ..
```

Business Parameter Descriptions

This section provides a technical description of the business parameters. You should familiarize yourself with the corresponding material in the Business Parameters section of the *Applications Manual*.

The following sections describe each Business Parameter. The material considers the application and technical issues, and the combination that will help you prepare for questions from the applications staff. Also remember that when you define a value for a particular Business Parameter, you are assigning the initial value that the program will use if it does not reset the value of the parameter itself.

Business Parameter # 1: HOLIDAY-TBL

This parameter specifies the holiday table that the invoking application programs will reference most often. The value must be alphanumeric, and under most circumstances this will refer to a table that contains holidays. If your shop does not use holidays at all, then this parameter is permitted to reference an empty table. The empty table must have a value for the TBL-ID and must be provided.

Business Parameter # 2: FISCYR-START

This parameter specifies the month that most invoking application programs will consider as the start of the fiscal year. The values range from 01 through 12.

If the fiscal year coincides with the calendar year, then this parameter should be set to the value of 01 (January), which states that the first fiscal month of the year and the first calendar month of the year are the same.

CA Calendar Routines uses this value when it performs a function related to the fiscal year, fiscal quarter, or fiscal month.

Business Parameter # 3: FISCMO-START

This parameter specifies the day of the month that most invoking application programs will consider as the start of the fiscal month. The values can range from 01 through 28.

If the fiscal month coincides with the calendar month, then this parameter should be set to the value of 01, which states that the first fiscal day of the month and the first calendar day of the month are the same.

CA Calendar Routines uses this value when it performs a function related to the fiscal year, fiscal quarter, or fiscal month.

Business Parameters # 2-3 (Combined): FISCAL-ID

This business parameter is a combination of the parameters #2: FISCYR-START and #3: FISCMO-START. Use this parameter for all Special Fiscal Calendar functions. It specifies which Special Fiscal Calendar table to access when processing Special Fiscal Calendar Functions (for example $444, $445, $454, or $455).

Business Parameter # 4: END-PNTS-DEF

This parameter specifies the manner in which CA Calendar Routines uses the FROM-DATE or the TO-DATE when performing various calculations.

There are four possible values:

- 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)

Note: This parameter is used explicitly by the following functions:

- 100 series
- 200 series except 207]
- 207 (parameter is set implicitly to T)
- 300 series
- 600 series
- 700 series
- 800 series (parameter is set implicitly to B)

Consider the following example; suppose that you 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 (for example, 15, 16, 17, 18)

If N is specified, the answer is 2 (for example, 16, 17)

If F is specified, the answer is 3 (For example, 15, 16, 17)

If T is specified, the answer is 3 (For example, 16, 17, 18)

The explanation is as follows. For the value B, both end points qualify, so there are four days (the two end points and the two days between). For the value N, neither end point qualifies, so there are two days (the two days between). For the value F, only the FROM-DATE qualifies, so there are three days (15 and the two days between). For the value T, only the TO-DATE qualifies, so there are three days (the two days between and 18).

Previously we considered an example of how END-PNTS-DEF affects the answer for a function that calculates the number of days between two dates. Now, let us consider an example of how END-PNTS-DEF affects the answer for a function that calculates a TO-DATE given a FROM-DATE and a direction/displacement. For example, suppose that we want to know the date that is three days from March 15. Again, there are four possibilities depending on the value of END-PNTS-DEF.

If B is specified, the answer is March 17 (for example, 15, 16, 17)

If N is specified, the answer is March 19 (for example, 16, 17, 18)

If F is specified, the answer is March 18 (for example, 15, 16, 17)

If T is specified, the answer is March 18 (for example, 16, 17, 18)

The explanation is as follows. For the value B, both end points qualify, so the three days are 15, 16, and 17. For the value N, neither end point qualifies, so the three days must lie between the end points of 15 and 19. For the value F, the FROM-DATE qualifies, so the three days are 15, 16, and 17; but since the TO-DATE, 18, does not qualify, it becomes the answer we seek. For the value T, the FROM-DATE does not qualify, so the three days are 16, 17, and 18, and since the TO-DATE, 18, does qualify, it remains our answer.

There are two more concepts that you should consider with the topic of End Points.

- If you choose B or N, you should be aware that CA Calendar Routines restricts certain operations. For example, you are not permitted to calculate the number of days between August 15, 2007 and August 15, 2007, if neither end point is to be included. Such unusual operations are described thoroughly in the appendix “Restricted Operations”.
- Most application programs could use a value of F or T to yield common sense answers for calendar day calculations. But, for processing day calculations, only T will yield common sense answers. Consider, as an example, the range 15,16,17,18 (in which 15 is a holiday). The number of calendar days for F and T is three (15-16-17 is the same 16-17-18). However, for processing days, a value of T will yield three days (16-17-18 all qualify) but a value of F will yield only two days (15-16-17, but 15 is disqualified).

Business Parameter # 5: DOW-STRING

This parameter lets you define the numerical values that correspond to each day of the week. It is composed of seven different settings. According to the CA Calendar Routines standard, position number 1 corresponds to Sunday, and position number 7 corresponds to Saturday. Any numerical value is acceptable, but repeat values are not allowed. Here are three sequences that you might like to consider.

- CA Calendar Routines 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

Business Parameter # 6: TRC-CONVR-PROC-DAY-DEF

This parameter defines the concept of processing day, or work day. It is actually composed of seven different settings; one for each day of the week. According to the CA Calendar Routines standard, position number 1 corresponds to Sunday, and position number 7 corresponds to Saturday. For each day of the week, you must choose one of the following values to specify inclusion or exclusion of holidays. As the system administrator, you should choose values that correspond to the concept of processing day for most invoking application programs.

- A = Always (A processing day, always)
- N = Never (A processing day, never)
- E = Exclude (A processing day, only if not a Holiday)
- I = Include (A processing day, only if a Holiday)

Suppose, for example, that most invoking application programs will consider processing days as weekday non-holidays. Then you should assign the value of NEEEEEN to the Business Parameter PROC-DAY-DEF.

As a second example, suppose that most invoking application programs consider processing days to be any weekday. In this case, the value of PROC-DAY-DEF should be NAAAAAN.

As a third example, if Sunday is never processed, Monday through Friday is always processed, and Saturday is processed (only if not a holiday), then the value of PROC-DAY-DEF should be NAAAAAE.

Business Parameter # 7: CENTURY-BREAK and SLIDING WINDOW Options

This parameter allows you to specify century dates for unknown centuries or to extend from the current year based on a specified slide range.

CENTURY-BREAK Option

This parameter allows you to specify the century for dates where the century is unknown (for example, YYMMDD). The Century Break syntax shown next allows you to specify the year below which one century value should be assigned and above which (or equal to) a different century should be assigned. In the syntax, AA is the lower century, BB is the upper century, and CC is the year where the break occurs. (The / and ; are required when specifying CENTURY-BREAK.)

AA/BB;CC

As an example, suppose the value of CENTURY-BREAK is 19/20;68. This means that if the year is 00 through 67, then the century is assumed to be 20; and, if the year is 68 through 99, then the century is assumed to be 19. As you can see this parameter lets you define what the century value should be, if it is missing from the date originally. This can be particularly useful if you need to use dates from files that have not been converted to include century information.

Note: CA Calendar Routines allows you to define a 100 year window anywhere in a 10,000 year range. Therefore, as an example, you can toggle between 18 and 19 for senior citizens.

SLIDING WINDOW Option

The Sliding Window option allows the window to extend from the current year based on the numeric *(nn*) value specified in the SLIDE-*nn* parameter. For example:

```
SLIDE-20
```

In this example, 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.

Business Parameter # 8: FROM-DATE-MASK

This parameter lets you define the format of the input FROM-DATE.

Note: There are several categories of masks.

Using Date Elements

You can describe the arrangement of various pieces of date information, such as year, month, and day. There are five arrangements: 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). (The applications programmer must 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). (The applications programmer must 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 (alpha-numeric). (The applications programmer must 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. (The applications programmer must 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

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

Note: 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 (2/18/95)
```

Special Null Mask

If you do not want multiple output date formats, use special Null Masks in Output Fields. This 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. (The applications programmer must 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 conventions exist for you to choose from. TDS is the CA Calendar Routines standard, S-1900G follows the 100 leap year rule, S-1900NG does not. (The applications programmer must 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 next 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)
```

As the System Administrator, you may find it helpful to know the editing criteria that are used with the specific date mask types.

Date Mask Types	Editing Criteria
Numeric Values	Numeric Values need leading zeros. Alphanumeric values need trailing spaces.
Gregorian	CCYY—0001 through 9999 YY—00 through 99 MM—01 through 12 DD—01 through max days per month
Julian	CCYY—0001 through 9999 YY—00 through 99 DDD—001 through 365/366
Single-Digit Century	H must be 8, 9, or 0. Designation R must be 0 or 1. S must be 1 or 2. T must be 0, 1, or 2.
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	Serial Dates must be numeric, within range and not correspond to February 29 of a non-leap year.

Business Parameters # 9,10,11,12 (TO-DATE-MASK, OUT1-DATE-MASK, OUT2-DATE-MASK, OUT3-DATE-MASK)

See Business Parameter # 8: FROM-DATE-MASK for more information.

Note: The product is delivered with OUT2 and OUT3 masks set to the null mask for efficiency.

Now that you have finished defining all of the business parameters, make sure the copybook TRCCONVR is still 1000 character positions long.

Business Parameter # 13: Validate-Null

This parameter defines the extent of input date validation performed by CA Calendar Routines (see TRC-CONVR-FROM-DATE and TRC-CONVR-TO-DATE) on standard Date Elements (see TRC-CONVR-FROM-DATE-MASK Using Date Elements). 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.

Business Parameter # 14: Set Alphabetic Output Case

This parameter sets the output alphabetic case for literals generated by CA Calendar Routines. Set this value to blank or 'u' for all uppercase literals or 'm' for mixed case literals (for example, January 1, 2007).

Chapter 3: Program Logic

This chapter describes the logic of the programs that comprise the core of CA Calendar Routines.

This chapter includes the following sections:

- How TRCHLGEN generates a holiday table
- How TRCHLGEN works

How TRCHLGEN Generates a Holiday Table

TRCHLGEN issues seven possible messages. All of them are marked on the flow diagram and described thoroughly in the appendix “Messages from TRCHLGEN”.

Step 1—Performs Preliminary Activities

TRCHLGEN begins with preliminary activities such as opening the files.

Step 2—Determines if Input File is Alphanumeric

The program reads through the input file. If TBL-ID is not alphanumeric, the program will issue MSG-000 and stop execution. Otherwise it will continue.

Step 3—Tests for Valid Dates

For each record, the program tests the input holiday to see if it is a valid date in CCYYMMDD format. If the date is not valid, the program will issue MSG-001.

Step 4—Adds Holiday to the Table

If the date is valid, the program will attempt to add the holiday to the table that is being built. If there is room in the table, the entry is added, otherwise the program will issue MSG-002.

Step 5—Sorts Holidays and Detects Duplicate Entries

After TRCHLGEN has read all of the holidays, it will sort the holidays that have been stored in the table, and then attempt to detect duplicate entries. This process will occur even if some input holidays were rejected, because the program wants to make you aware of duplicate entries. This will let you decide if the duplicate should have been keyed as an entirely different date. If a duplicate is detected, the program will issue MSG-003.

Step 6—Determines if There Was Any Valid Input

At the beginning of the next phase of processing, the program asks itself two important questions. “Was there any input?,” and (if there was any input), “Were there any errors?” The answers will determine whether the program generates a copy/prog/flatinfo-record and/or calendar listings. There are only three possibilities, each with an associated message number.

- MSG-004: There was no input. The program will generate a copy/prog/flatinfo-record representing an empty table. However the program will not generate calendar listings.
- MSG-005: There was input, and there were no errors. The program will generate a copy/ prog/flatinfo-record containing all the holidays that you requested. Additionally, the program will generate calendar listings.
- MSG-006: There was input, but there were errors. The program will not generate a copy/ prog/flatinfo-record, and it will not generate calendar listings.

Step 7—Closes the Files

The program ends its processing with tasks such as closing the files.

How TRCENGIN Works

The CICS version of this program is TRCENGNC.

Step 1—Editing the Input Parameters

TRCENGIN begins with preliminary activities such as editing the input parameters. Also, if warranted, the appropriate holiday table and/or Fiscal Calendar table is loaded.

TRCSMALL is a functionally limited version of TRCENGIN, follows the basic flow of this program, and is limited to the following functions:

Code	Routines
C101	DAYS BETWEEN
C102	DATE +/- DAYS
C103	DATE +/- WEEKS, MONTHS, YEARS
C201	Get CALENDAR NEXT DAY
C202	Get CALENDAR PREVIOUS DAY
U001	Get CURRENT DATE

Code	Routines
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

Step 2—Performs Functions

Within the main processing there are two major categories of functions:

- Utilities, which are highly specialized and treated individually.
- A collection of functions that perform calendar day calculations and processing day calculations.
- This collection of functions is divided into five different groups; which are:
  - Basic
  - Extended
  - Identify
  - Since/Until
  - Relative To

  Note: The 800 series (Nth DOW Occurrence Within a Period), is simply implemented as a variation of the “Relative To” series of functions.

Utility Functions

The Utility functions perform retrieval of current date and time information, editing, reformatting, and simple comparison of date values. The Utility functions are:

Utility	Description
(U001)	CURRENT DATE-TIME
(U002)	VALIDATE DATE
(U003)	LEAP YEAR
(U004)	HOLIDAY

Utility	Description
(U005)	PROCESSING DAY
(U006)	DAY OF WEEK
(U007)	DATE -> ABSOLUTE DAYS
(U008)	ABSOLUTE DAYS --> DATE
(U009)	REFORMAT DATE
(U010)	AGE FUNCTION
(U011)	COMPARISON FUNCTION
(U012)	CONCATENATE THE CENTURY FUNCTION
(U013)	TRUNCATE THE CENTURY FUNCTION

Basic Functions

First let us consider the two Basic functions:

- Days Between—Calculates the number of days from the FROM-DATE to the TO-DATE.
- Date +/- N Days—Determines the TO-DATE given the FROM-DATE and a Direction/ Displacement.

These two basic functions are referred to as primitive routines, and are labeled (1) and (2) within the flow diagram. They are also the points in the processing where the TRCENGIN tests for restricted operations. If the operation is not restricted, then the next major decision is made.

If the request is for *calendar days,* then simple arithmetic calculations are performed; otherwise, the request must be for *processing days,* in which case the March Along routine is invoked. The March Along routine is a dual purpose routine that is used to travel up or down an imaginary calendar, seeking a date or counting intervening days.

Extended, Identify, Since/Until, and Relative To Functions

Next, consider the four remaining groups of functions,

- Extended
- Identify
- Since/Until
- Relative To

There are two important things to notice, in this regard:

- All four groups invoke a primitive routine for Date Identification. This primitive routine is the point where TRCENGIN adjusts a date forward or backward if only processing days are being considered.
- The groups Extended, Since/Until, and Relative To all invoke the primitives Days Between and Date +/- N Days.

Step 3—Formats the Output

The program ends with tasks such as formatting the output.

Appendix A: Messages from TRCHLGEN

This appendix provides a brief description of the messages that may appear from the program that generates a holiday table.

Messages 000 - 006

000

TBL-ID Should be Alphanumeric

Reason:

The TBL-ID specified is not 00 through ZZ.

Action:

Change the TBL-ID to an alphanumeric value of 00 through ZZ.

001

This is not a valid date in CCYYMMDD Format

Reason:

The date in the input HOLIDAY CCYYMMDD Format file is not a valid date in CCYYMMDD format.

Action:

You must correct the date so that it is a valid date in CCYYMMDD format.

002

Too many holidays, increase table size

Reason:

Too many holiday dates have been included in the input HOLIDAY file.

Action:

Follow this procedure:

1. In copybook TRCCOMNW, increase TRC-HLRTV-DTL-ENTRY.
2. In program TRCHLGEN:
    a. Increase TRC-HLGEN-ENTRY-DTL-MAX
    b. Increase TRC-HLGEN-ENTRY-ALL
    c. Increase the record length for the FLATINFO file
        1. Block contains
        2. Record contains
        3. 01 record definition
3. In program TRCENGIN, increase TRC-HLTBL-DTL-ENTRY
4. Recompile and relink TRCHLGEN and TRCENGIN.

003

This date has already appeared

Reason:

You have supplied a date in the input HOLIDAY file that appears more than once.

Action:

You must remove all repeat occurrences of the date that caused this error message to appear.

004

No Input;Calendars-No,Copy/ Prog/Flatinfo-Yes

Reason:

You did not supply any input to the program. This is acceptable if your intention is to generate a NULL Holiday Table. In that case the program will only generate a copybook/ program/flatinfo-record, and they will represent an empty holiday table.

Action:

If you thought that holidays were going to be read in, then you must research why the input file was empty.

005

Good Input;Calendars-Yes,Copy/Prog/Flatinfo-Yes

Reason:

This is the result of normal processing. It means that the program will generate a calendar listing (with holidays marked) from January 1 of the earliest year through December 31 of the latest year. It also means the program will create a copybook/program/flatinfo-record containing every holiday you supplied.

Action:

Informational, no action required.

006

Bad Input;Calendars-No,Copy/Prog/Flatinfo-No

Reason:

This means that the input holiday file contains errors. As a result, the program will not generate calendars or a copybook/program/flatinfo-record.

Action:

You must correct the errors that printed earlier in the output listing. These errors identified invalid and/or duplicate holidays

Appendix B: SYSERRS from TRCENGIN

This appendix provides a brief description of each system error that can appear from the engine program TRCENGIN.

There are several system level messages that can be generated from program TRCENGIN. These messages will occur if the program has been corrupted. If you have made modifications to TRCENGIN and one of these messages begins to appear, then use this appendix to interpret the nature of the corruption. It will help you during debugging.

Messages 900 - 994

900

Reason:

The Function Type is “C” or “P”, but the Function Number is not in the allowable range of “C” or “P” type functions.

901

Reason:

An unknown FLEXIO token type has slipped through.

902

Reason:

During the editing of the IN-NUMERIC-PARM, a FUNCTION-TYPE was discovered which was not an end-type or a begin-type.

903

Reason:

The Function Type is not “C”, “P”, “U”, or “K”.

904

Reason:

The Function Type is “U”, but the Function Number is not a recognized Utility Functions (e.g. “CURRDATE”, “VALIDATE”, “LEAP-YEAR”, “HOLIDAY”, “PROC-DAY”, “DOW”, “DATE-TO-ABS”, “ABS-TO-DATE”, “REFORMAT”, or “AGE”).

905

Reason:

The Function Number is in the Range “BASIC”, but the Function Number is not an allowed “BASIC” Function. (e.g. “BETWEEN” or “DATE +/- N Days” or DATE +/- N Weeks/Months/Years).

906

Reason:

The Function Number is in the Range “EXTENDED”, but the Function Number is not an allowed “EXTENDED” Function. (e.g. “NEXT DAY”, “PREV DAY”, “Days Since Beginning of the Month”, “Days Until the End of the Month”, “Beginning of the Month +/- N Days”, “End of the Month +/- N Days”, “Nth DOW Occurrence Forward/Backward”).

907

Reason:

The Function Number is not in one of the three ranges that specifies the Format of the Output. The output Format must be specified for all Functions, and must be either (1) a format that specifies a Date on the 365-Calendar, (2) a format that specifies a Date on the 360-Calendar, or (3) a format that specifies a NUMERIC-PARM.

908

Reason:

The Program has referenced a current date that is not valid. Make sure the logic and method of retrieval provide an accurate current date in CCYYMMDD format.

909

Reason:

The Internal Date Format is not type “GREGORIAN”, “JULIAN”, or “Calendar-360”.

910

Reason:

The Function Number is not type “MONTH, “QUARTER”, or “YEAR”.

911

Reason:

The program is in “QUARTER” Processing, and the Function Number is not type “BEGINNING” or “END”.

912

Reason:

The program is in “YEAR” Processing and the Function Number is not type “BEGINNING” or “END”.

913

Reason:

The Program is in “NEXT” Processing, and the Function Number is not type “MONTH”, “QUARTER”, or “YEAR”.

914

Reason:

The Program is in “PREV” Processing, and the Function Number is not type “MONTH”, “QUARTER”, or “YEAR”.

915

Reason:

The Function Number is not type “CURR”, “NEXT” or “PREV”.

916

Reason:

The Program has failed internally while attempting to determine a Future Date.

917

Reason:

The Program has failed internally while attempting to determine a Past Date.

918

Reason:

The Program wants to define the Day of the Month, but the Function Number is not type “BEGINNING” or “END”.

919

Reason:

The Program wants to perform a “Forward Adjustment” from the Beginning of the Month; or perform a “Backward Adjustment” from the “End of the Month” --- but the Function Number is not type “BEGINNING” or “END”.

920

Reason:

The Program wants to perform a “BETWEEN” Function, but the Function Type is not “C” or “P”.

921

Reason:

The Program wants to perform a “DATE +/- N Days” Function, but the Function Type is not “C” or “P”.

922

Reason:

For a Non-Calendar-360 computation, the END-PNTS-DEF specifies “BOTH”, and the program has allowed the situation where FROM-DATE = TO-DATE, to slip through.

923

Reason:

For a Non-Calendar-360 computation, the END-PNTS-DEF specifies “NEITHER”, and the program has allowed the situation where FROM-DATE = TO-DATE, to slip through.

924

Reason:

For a Non-Calendar-360 computation, the Program is processing a “BETWEEN” function and the END-PNTS-DEF is not “B”, “N”, “F”, or “T”.

925

Reason:

For a Non-Calendar-360 computation, the END-PNTS-DEF specifies “BOTH”, and the program has allowed the situation where the “Direction/ Displacement” is zero.

926

Reason:

For a Non-Calendar-360 computation, the END-PNTS-DEF specifies “NEITHER”, and the program has allowed the situation where the “Direction/Displacement” is zero.

927

Reason:

For a Non-Calendar-360 computation, the Program is processing a “DATE +/ - N Days” function and the END-PNTS-DEF is not “B”, “N”, “F”, or “T”.

928

Reason:

The Program is processing a lower level “BETWEEN” Function, and the FROM-DATE = TO-DATE situation has been discovered.

929

Reason:

The Program is processing a lower level “DATE +/- N Days” Function, and the Direction/Displacement = 0, situation has been discovered.

930

Reason:

A holiday entry has been discovered, and the program must decide whether the next holiday of interest is in the future or the past; but the appropriate indicator does not specify “Forward” or “Backward”.

931

Reason:

The routine that “marches along”, does not know whether to increment or decrement the Date, because the appropriate indicator does not specify “Forward” or “Backward”.

932

Reason:

The routine that “marches along”, does not know whether to increment or decrement the Day of the Week, because the appropriate indicator does not specify “Forward” or “Backward”.

933

Reason:

The PROCESSING-DAY-DEFINITION, for one of the days of the week, is not “A”, “N”, “E”, or “I”.

934

Reason:

The routine that initially decides whether we want to try to match holidays toward the future or the past, cannot do so, because the appropriate indicator does not specify “Forward” or “Backward”.

935

Reason:

The Detail Holiday Table pointer is greater than the maximum.

936

Reason:

The Detail Holiday Table pointer is less than 1.

937

Reason:

A FUNCTION-TYPE other than Calendar or Fiscal has been encountered.

938

Reason:

While adjusting the day, a FUNCTION-TYPE was encountered other than a Begin-type or End-type.

939

Reason:

While adjusting the year, a FUNCTION-TYPE was encountered other than a Begin-Type or End-Type.

940

Reason:

While processing a Begin-Type function, a month other than 1-12 was encountered when the FISCYR-START is 1, 4, 7, or 10.

941

Reason:

While processing a Begin-Type function, a month other than 1-12 was encountered when the FISCYR-START is 2, 5, 8, or 11.

942

Reason:

While processing a Begin-Type function, a month other than 1-12 was encountered when the FISCYR-START is 3, 6, 9, or 12.

943

Reason:

While processing a Begin-Type function, the program discovered a FISCYR-START other than 1-12.

944

Reason:

While processing a End-Type function, a month other than 1-12 was encountered when the FISCYR-START is 1, 4, 7, or 10.

945

Reason:

While processing a End-Type function, a month other than 1-12 was encountered when the FISCYR-START is 2, 5, 8, or 11.

946

Reason:

While processing a End-Type function, a month other than 1-12 was encountered when the FISCYR-START is 3, 6, 9, or 12.

947

Reason:

While processing a End-Type function, the program discovered a FISCYR-START other than 1-12.

948

Reason:

While trying to determine Fiscal Quarter end points, a Function Number was discovered other than Begin-Type or End-Type.

949

Reason:

While trying to define the nominal end points for fiscal quarter processing, a Function Number was discovered other than Begin-Type or End-Type.

950

Reason:

While trying to massage the function number to determine the period-end, a Function Number was discovered other than Begin-Type or End-Type.

951

Reason:

While trying to massage the function number back to its original value, after determining the period-end, a Function Number was discovered other than Begin-Type or End-Type.

952

Reason:

While trying to determine if the derived date is beyond the period start or end, a Function Number was discovered other than Begin-Type or End-Type.

953

Reason:

An unknown Flexio token type (space, zero, delimiter) has slipped through on input.

954

Reason:

Upon date implosion, an attribute type other than alpha and numeric has been detected.

955

Reason:

Upon date explosion, an attribute type other than alpha and numeric has been detected.

956

Reason:

An unknown Flexio token type (space, zero, delimiter) has slipped through on output.

957

Reason:

While processing the AGE Function, an IN-ALPHA-PARM of other than ENFORCE or NO-ENF has slipped through.

958

Reason:

While processing the AGE Function, the program has set the IN-ALPHA-PARM to something other than Y-ENF or Y-NOENF.

959

Reason:

The IN-ALPHA-PARM should be of the type Y/N (i.e. “YES” or “NO”) or of the type WMY (“WEEK”, “MONTH” or “YEAR”).

960

Reason:

For Function 103, the duration specified by IN-ALPHA-PARM is not Weeks, Months or Years.

961

Reason:

Upon an attempt to progress Forward or Backward “N” months, it is discovered that the number of months is zero.

962

Reason:

Upon an attempt to correct Forward or Backward for a holiday, it is discovered that the original displacement was zero.

963

Reason:

The Function is not Calendar-360 or Non-Calendar-360.

964

Reason:

The Function Group is Calendar-360, but the function number is not one of the recognized Calendar-360 functions.

965

Reason:

The Table containing the Calendar-360 / Calendar-365 Information is missing a Calendar-360 entry (i.e. 1-360).

966

Reason:

The Table Containing the Calendar-360 / Calendar-365 Information is missing a Calendar-365 entry (i.e. Jan 01 through Dec 31).

967

Reason:

For a Calendar-360 computation, the END-PNTS-DEF specifies “BOTH”, and the program has allowed the situation where FROM-DATE = TO-DATE, to slip through.

968

Reason:

For a Calendar-360 computation, the END-PNTS-DEF specifies “NEITHER”, and the program has allowed the situation where FROM-DATE = TO-DATE, to slip through.

969

Reason:

For a Calendar-360 computation, the Program is processing a “BETWEEN” function and the END-PNTS-DEF is not “B”, “N”, “F”, or “T”.

970

Reason:

For the Calendar-360 computation Date +/- N, the duration is not Day, Month, or Year.

971

Reason:

For a Calendar-360 computation, the END-PNTS-DEF specifies “BOTH”, and the program has allowed the situation where the “Direction/Displacement” is zero.

972

For a Calendar-360 computation, the END-PNTS-DEF specifies “NEITHER”, and the program has allowed the situation where the “Direction/ Displacement” is zero.

973

Reason:

For a Calendar-360 computation, the Program is processing a “DATE +/- N Days” function and the END-PNTS-DEF is not “B”, “N”, “F”, or “T”.

974

Reason:

A FROM-DATE-MASK other than edited or non-edited has slipped into the system.

975

Reason:

A TO-DATE-MASK other than edited or non-edited has slipped into the system.

976

Reason:

A OUT1-DATE-MASK other than edited or non-edited has slipped into the system.

977

Reason:

A OUT2-DATE-MASK other than edited or non-edited has slipped into the system.

978

Reason:

A OUT3-DATE-MASK other than edited or non-edited has slipped into the system.

979

Reason:

While processing the AGE Function, the program has advanced past the TO-DATE, even though it was traveling one day at a time. This is detected at the lower level routine.

980

Reason:

While processing the AGE Function, the program has passed the TO-DATE during the jump-by-years cycle.

981

Reason:

While processing the AGE Function, the program has advanced past the TO-DATE, while traveling one month at a time.

982

Reason:

While processing the AGE Function, the program has advanced past the TO-DATE, even though it was traveling only one day at a time.

983

Reason:

The INTERNAL-DATE-MASK is SERIAL-TYPE, but is not one of the allowed SERIAL-DATE designations.

984

Reason:

While processing the AGE Function, an END-POINTS-DEF of something other than “B”, “N”, “F” or “T” was detected.

985

Reason:

A FLEXIO MASK TYPE other than Input or Output was detected.

986

Reason:

Upon input, an unknown FLEXIO Conversion Type was discovered.

987

Reason:

Upon output, an unknown SERIAL Date Type has been discovered.

988

Reason:

The Program has referenced a current time that is not valid. Make sure the logic and method of retrieval provide an accurate current time in HHMMSSHH format.

989

Reason:

Invalid Day of Week value—must be 1-7

990

Reason:

Invalid Extended function code—must be 5nn, 6nn, 7nn, or 8nn.

991

Reason:

Invalid function code—beyond 8nn.

992

Reason:

Function code U011 (Quick Compare) Date Mask invalid—must be GENR-X, CCYYMMDD, YYMMDD--, --YYMMDD, YYDDD---, ---YYDDD, YYMM----, ----YYMM, YY------, ------YY.

993

Reason:

From and To date masks are not equal.

994

Reason:

Settings for TRC-CONVR-VALIDATE-NULL and TRC-ENHNC-EDIT-NULL-SW incorrectly set. See Null Character Date Mask Validation.

Appendix C: Restricted Operations

This appendix provides a brief description of the operations that CA Calendar Routines will not allow.

- Days Between and Date +/- N Days Operations
- DAYS BETWEEN
- DATE +/- N DAYS
- Correction Factors
- Special Cases of From-Date = To-Date and DD=

Days Between and Date +/- N Days Operations

CA Calendar Routines maintains symmetry between two types of operations.

DAYS BETWEEN

For this type of operation, we know the values of FROM-DATE and TO-DATE. The value we seek is the Direction/Displacement. These values are related to one another, nominally, by the following formula:

DD = T–F

where:

- F = FROM-DATE
- T = TO-DATE
- DD = Direction/Displacement

DATE +/- N DAYS

For this type of operation, we know the values of FROM-DATE and Direction/ Displacement. The value we seek is the TO-DATE. These values are related to one another, nominally, by the following formula (using the same values described above):

T = F+DD

Correction Factors

It is important to note the use of the word *nominally* in the previous sections. It refers to the relationship between FROM-DATE, TO-DATE, and Direction/Displacement, only when TRC-CONVR-END-PNTS-DEF is set to F or T, representing (FROM-DATE, only or TO-DATE, only). Otherwise, for the settings of B (BOTH) and N (NEITHER), correction factors must be supplied in the formulas shown in the previous sections.

Furthermore, the correction factors are dependent on other considerations. These extra concerns are summarized in the formula shown next.

```
"DAYS BETWEEN"                          "DATE +/- N DAYS"
 F>T                                     DD<0
    "B" DD = T - F - 1 (1A)                 "B" T = F + DD + 1 (2A)
    "T" DD = T - F (1B)                     "T" T = F + DD (2B)
    "F" DD = T - F (1C)                     "F" T = F + DD (2C)
    "N" DD = T - F + 1 (1D)                 "N" T = F + DD - 1 (2D)
 F<T                                     DD>0
    "B" DD = T - F + 1 (1I)                 "B" T = F + DD - 1 (2I)
    "T" DD = T - F (1J)                     "T" T = F + DD (2J)
    "F" DD = T - F (1K)                     "F" T = F + DD (2K)
    "N" DD = T - F - 1 (1L)                 "N" T = F + DD + 1 (2L)
```

Special Cases of From-Date = To-Date and DD=0

Now we must consider the special cases of From-Date = To-Date and DD=0. As we can see from the following examples, ambiguous results can occur for these special conditions when TRC-CONVR-END-PNTS is B (Both) or N (Neither). As a matter of policy, CA Calendar Routines just restricts such operations rather than assigning an arbitrary answer such as zero.

```
"DAYS BETWEEN"
END-POINTS FROM TO FORMULA DD COMMENTS FORMULA
"B" 19 17 DD = T - F - 1 -3 (1A)
"T" 19 17 DD = T - F -2 (1B)
"F" 19 17 DD = T - F -2 (1C)
"N" 19 17 DD = T - F + 1 -1 (1D)
"B" 19 18 DD = T - F - 1 -2 (1A)
"T" 19 18 DD = T - F -1 (1B)
"F" 19 18 DD = T - F -1 (1C)
"N" 19 18 DD = T - F + 1 -0 (1D)
"B" 19 19 DD = T - F +/- 1 +/- 1 Restricted (1E)
"T" 19 19 DD = T - F 0 (1F)
"F" 19 19 DD = T - F 0 (1G)
"N" 19 19 DD = T - F -/+ 1 -/+ 1 Restricted (1H)
"B" 19 20 DD = T - F + 1 2 (1I)
"T" 19 20 DD = T - F 1 (1J)
"F" 19 20 DD = T - F 1 (1K)
"N" 19 20 DD = T - F - 1 0 (1L)
"B" 19 21 DD = T - F + 1 3 (1I)
"T" 19 21 DD = T - F 2 (1J)
"F" 19 21 DD = T - F 2 (1K)
"N" 19 21 DD = T - F - 1 1 (1L)
"DATE +/- N DAYS"
END-POINTS FROM DD FORMULA T COMMENTS FORMULA
"B" 19 -2 T = F + DD + 1 18 (2A)
"T" 19 -2 T = F + DD 17 (2B)
"F" 19 -2 T = F + DD 17 (2C)
"N" 19 -2 T = F + DD - 1 16 (2D)
"B" 19 -1 T = F + DD + 1 19 (2A)
"T" 19 -1 T = F + DD 18 (2B)
"F" 19 -1 T = F + DD 18 (2C)
"N" 19 -1 T = F + DD - 1 17 (2D)
"B" 19 0 T = F + DD +/- 1 20,18 Restricted (2E)
"T" 19 0 T = F + DD 19 (2F)
"F" 19 0 T = F + DD 19 (2G)
"N" 19 0 T = F + DD -/+ 1 18,20 Restricted (2H)
"B" 19 1 T = F + DD - 1 19 (2I)
"T" 19 1 T = F + DD 20 (2J)
"F" 19 1 T = F + DD 20 (2K)
"N" 19 1 T = F + DD + 1 21 (2L)
"B" 19 2 T = F + DD - 1 20 (2I)
"T" 19 2 T = F + DD 21 (2J)
"F" 19 2 T = F + DD 21 (2K)
"N" 19 2 T = F + DD + 1 22 (2L)
```

Appendix D: Mnemonics

This appendix gives you an alphabetical listing of the mnemonics used throughout CA Calendar Routines source code.

Naming Conventions

Variable names within CA Calendar Routines follow a simple naming convention:

- The first four characters are always TRC-.
- The next five characters provide an abbreviated description of the purpose or usage of the variable.

The following table describes the meanings of all mnemonics used throughout CA Calendar Routines.

Mnemonic	Meaning
ABSDN	Absolute Down
ABSUP	Absolute Up
ABTJL	Absolute to Julian
ABTDW	Absolute to Day of Week
ADJST	Adjust forward or backward
BTWEN	Between
CONVR	Conversational
CPYBK	Copybook
CURDT	Current Date
DAPOM	Date Plus or Minus
DOWUP	Day of Week Up
DOWDN	Day of Week Down
ENGIN	Engine
ENHNC	Enhance
ERRMS	Error Messages
EXTRN	External Module

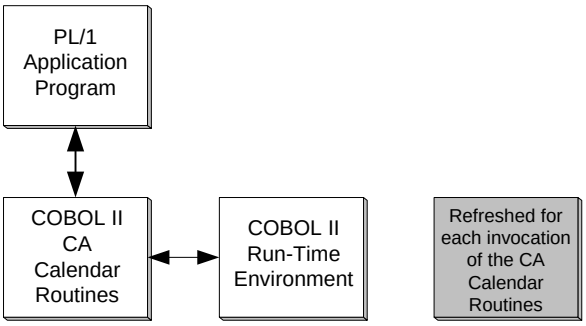
Mnemonic	Meaning
FLXIO	Flexible Input/Output
JULDN	Julian Down
JULUP	Julian Up
JLTGR	Julian to Gregorian
JLTAB	Julian to Absolute
INSTL	Installation
GRTJL	Gregorian to Julian
GRGDN	Gregorian Down
GRGUP	Gregorian Up
HLGEN	Holiday Generation
HLTBL	Holiday Table
HLRTV	Holiday Retrieval
IDENT	Identification
INTRL	Internal
LPYER	Leap Year
MARCH	March Along
NTHDW	Nth Day of the Week
PPCAL	Perpetual Calendar
QUALF	Qualification
RTNCD	Return Code
SETHL	Set Holiday
SYSCN	System Constants
VALAB	Validate Absolute Days
VALDW	Validate Day of Week
VALGR	Validate Gregorian Date
VALJU	Validate Julian Date
YRNFO	Year Information

Appendix E: IBM MVS PL/1 and Assembler Program CALLs to CA Calendar Routines

CA Calendar Routines is written in COBOL based on the ANSI '85 COBOL language standard. This appendix applies to installations that have compiled CA Calendar Routines using an MVS Cobol compiler (such as COBOL II or VS-Cobol). The MVS compilers include a COBOL run-time module with the CA Calendar Routines executable module. This module is part of IBM's Language Environment.

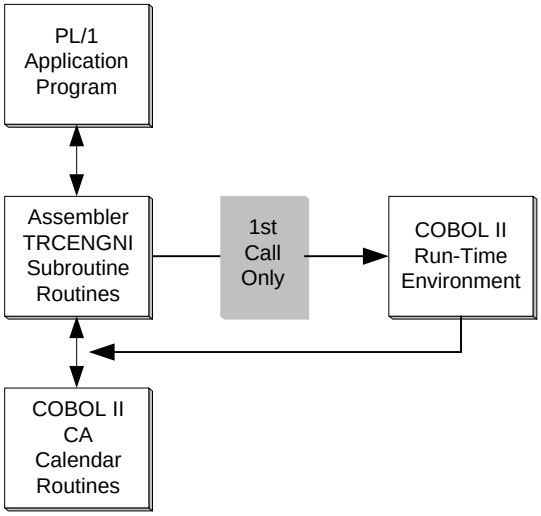
Overview

A problem occurs whenever a non-Cobol program CALLs the CA Calendar Routines:



Refreshing the Cobol II Run-time environment is NOT required and causes significant performance degradation. The COBOL II Run-time environment should ONLY be loaded during the initial execution of the Calendar Routines.

A subroutine has been developed to correct multiple refreshment of the Cobol II Run-time environment. This subroutine is called instead of the CA Calendar Routines. The subroutine loads the Cobol II Run-time during the initial invocation of the subroutine, invokes the Cobol II Calendar Routines, and leaves the Run-time environment intact for future invocations.



Three files should have been previously loaded into the source statement file during the first steps of CA Calendar Routines installation (see the chapter “Installation”).

TRCENGNI

TRCENGNI is Assembler source code. Your PL/I or Assembler application program will call this program instead of the Cobol II TRCENGIN Calendar Routines program. This program controls the Inter Language Call environment, loading of the Cobol II Run-time environment, and Calls to TRCENGIN.

```
TITLE 'TRCENGNI - COBOL II INTERFACE ROUTINE'
*************************************************************************
* FUNCTION: ESTABLISH A REUSABLE COBOL II RUNTIME ENVIRONMENT ON
*   THE FIRST INVOCATION
*
*HOW INVOKED: CALL TRCENGNI( <PASS-THROUGH PARMS> )
*
*SPECIAL CONSIDERATIONS:
*  SEE COBOL II PROGRAMMER'S GUIDE FOR USE AND RESTRICTIONS OF
*  OF USING IGZERRE.
*************************************************************************
SPACE 3
************ * * * DESCRIPTION OF PROGRAM BEHAVIOR * * * ************
*
* INITIALIZATION:
*  NONE.
*
*
* GENERAL:
```

```
*
*********************************************************************
SPACE 3
**************** * * * C H A N G E L O G  * * * ****************
*
*
* __DATE__ _____________________ WHAT AND WHO _____________________
*
* 10/22/98 ORIGINAL CODE - ADPAC CORPORATION - JLG
* 12/23/98 DELETED MACRO FOR RUN DATE – LWL
*********************************************************************
 TITLE 'TRCENGNI MAINLINE - ENTRY CODE'
*********************************************************************
* REGISTER USAGE
*  R0 - R3 = SCRATCH
* R4 = INTERNAL LINKAGE REGISTER
* R6 = PTR TO I/P AND O/P CONVERSION FACTOR TABLES
* R7 = BASE REG FOR THE CALLING PARM STRUCTURE
*  R11 = SINGLE BASE REGISTER
*********************************************************************
SPACE 3
PRINT NOGEN
SPLEVEL SET=2
R0 EQU 0.
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12   *** NOT MODIFIED - FOR PL/1 ENVIRONMENT ***
R13 EQU 13
R14 EQU 14
R15 EQU 15
DC C'TRCENGNI'
DC AL1(7)
ENTRY TRCENGNI
TRCENGNI CSECT
TRCENGNI AMODE 31
TRCENGNI RMODE ANY
USING TRCENGNI,R15
B MAINLINE
IDENT  DC CL8'TRCENGNI'
DC CL8'V 1.00.0'
CDATEH *** ASSEMBLY DATE YYYY/MM/DD ***
DC C' - PLATINUM CA CALENDAR ROUTINES'
TITLE 'TRCENGNI MAINLINE'
*********************************************************************
*REGISTER USAGE:
*********************************************************************
MAINLINE DS  0H
STM R14,R12,12(R13)   SAVE CALLER'S REGS
LA R11,SAVEAREA
ST R11,8(R13)  POINT CALLER'S SA TO THIS ONE
ST R13,4(R11)
LR R13,R11    ESTABLISH NEW SA
USING  SAVEAREA,R13
BC NOP DO_CALL    *** MODIFIED INSTRUCTION! ***
MVI BC+1,X'F0'  CHANGE TO UNCONDITIONAL BRANCH
LA R1,1  INITIALIZE CALL
L R15,=V(IGZERRE)
DROP R15
BASR R14,R15    ESTABLISH COBOL II RUNTIME ENV
DO_CALL DS 0H
BALR R11,0
USING *,R11
L R1,SAVEAREA+4    R1=@OF CALLER'S SA
L R1,24(R1)   RELOAD CALLER'S R1
L R15,=V(TRCENGIN)
```

```
BASR R14,R15
GOODEXIT DS  0H
L R13,4(R13)
L R14,12(R13)
LM R0,R11,20(R13)    RESTORE ALL BUT R15
BR R14  RETURN
SAVEAREA DC  9D'0'
LTORG
END
```

ASMHJCL

ASMHJCL is the JCL to Assemble TRCENGINI. You will need to modify the job card and the library names. Run this once to create load module TRCENGINI.

```
//tsoid JOB (999,99999),'TEST ASSEMBLY',CLASS=A,
// MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=tsoid
//*
/*JOBPARM L=20,T=5,C=1000,ROOM=SNFR,N=1,K=0,SYSAFF=*
//*
//ASMIT EXEC ASMHCL,REGION=2M,TIME=(,15),
// PARM.C=('OBJECT,XREF(SHORT),NORENT,LINECOUNT(60)',
// 'USING(NOLIMIT,MAP,WARN(0)),ASA,NOALIGN')
//C.SYSLIB DD DSN=your.xxx.SOURCE,DISP=SHR
// DD DSN=your.xxx.COPYLIB,DISP=SHR
// DD DSN=SYS1.MACLIB,DISP=SHR
/ DD DSN=SYS1.MODGEN,DISP=SHR
//C.SYSIN  DD DISP=SHR,DSN=your.xxx.SOURCE(TRCENGNI)
//L.SYSLMOD DD DISP=SHR,DSN=your.xxx.LOADLIB
//L.SYSIN  DD *
NAME TRCENGNI(R)
//L.SYSLIB DD DSN=SYS1.COB2LIB,DISP=SHR
//  DD DSN=SYS1.COB2COMP,DISP=SHR
//  DD DSN=your.xxx.LOADLIB,DISP=SHR
```

PLICOMP

PLICOMP is sample JCL to compile your PL/I application, with it's call to TRCENGNI. Assembler users treat a call to TRCENGNI just like any other Assembler call.

```
//tsoid  JOB (999,9999),'PLI TCCR COMPILE',MSGCLASS=X,CLASS=A,
//  MSGLEVEL=1,NOTIFY=tsoid
//*
//JOBLIB DD DSN=PLI.V2R3M0.PLIBASE,DISP=SHR  << PL/I LIBRARIES
// DD DSN=PLI.V2R3M0.SIBMBASE,DISP=SHR
//*
//COMP1  EXEC PGM=IEL0AA,
//*****************************************************************
//  PARM=('AG,GS,NEST,OPT(2),X(S),MACRO')
//STEPLIB DD DSN=PLI.V2R3M0.PLICOMP,DISP=SHR  << COMPILER
//SYSLIB  DD DISP=SHR,DSN=your.xxx.copylib   << COPYLIB
//SYSIN  DD DISP=SHR,DSN=your.xxx.sourcelib(progname) << SOURCE
//SYSPRINT DD SYSOUT=*,HOLD=YES
//SYSLIN  DD DSN=&&COMP1,
//   DISP=(NEW,PASS),
//   UNIT=SYSDA,
//   SPACE=(TRK,(5,5),RLSE),
//   DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSUT1  DD UNIT=SYSDA,SPACE=(1024,(200,50),,CONTIG,ROUND),
//   DCB=BLKSIZE=1024
//*
//LKED1  EXEC PGM=IEWL,COND=(5,LT,COMP1),PARM='LET,MAP'
//SYSLIB  DD DSN=your.xxx.LOADLIB,DISP=SHR   << YOUR LOADLIB
//  DD DSN=PLI.V2R3M0.PLIBASE,DISP=SHR  << PL/I LIBS
//  DD DSN=PLI.V2R3M0.SIBMBASE,DISP=SHR
//  DD DSN=SYS1.COB2LIB,DISP=SHR  << COBOLII LIBS
//  DD DSN=SYS1.COB2COMP,DISP=SHR
//*
//SYSLIN  DD DSN=&&COMP1,DISP=(OLD,DELETE)
//SYSLMOD DD DISP=SHR,DSN=your.xxx.LOADLIB(progname) << LOADLIB
//SYSUT1  DD UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD)),
//   SPACE=(1024,(50,20),RLSE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NAME progname(R)
```

Index

N

P

S

T