CA-MetaCOBOL™ +

Quality Assurance Guide

Release 1.1

R203M+11DRP

-- PROPRIETARY AND CONFIDENTIAL INFORMATION --

This material contains, and is part of a computer software program which is, proprietary and confidential information owned by Computer Associates International, Inc. The program, including this material, may not be duplicated, disclosed or reproduced in whole or in part for any purpose without the express written authorization of Computer Associates. All authorized reproductions must be marked with this legend.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the United States Government ("the Government") is subject to restrictions as set forth in A) subparagraph (c) (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, and/or B) subparagraphs (c) (1) (ii) of the Rights in Technical Data and Computer Software clause of DFAR 252.227-7013. This software is distributed to the Government by:

Computer Associates International, Inc.
One Computer Associates Plaza
Islandia, New York 11788-7000

Unpublished copyrighted work - all rights reserved under the copyright laws of the United States.

This material may be reproduced by or for the United States Government pursuant to the copyright license under the clause at DFAR 252.227-7013 (October 1988).

Release 1.1, January, 1992
Updated March, 1992

Copyright © 1992 Computer Associates International, Inc.
One Computer Associates Plaza, Islandia, NY 11788-7000
All rights reserved.

CA-LIBRARIAN® is a registered trademark of Computer Associates, Inc. CA-MetaCOBOL™ + and CA-DATADICTIONARY™ are trademarks of Computer Associates, Inc

All product names referenced herein are trademarks of their respective companies

Contents

CA-MetaCOBOL+
Technical Updates
Release 1.1

IM

Quality Assurance Guide
March 1992

INSERT (I)	new pages wherever you see (I) next to the pages listed below.
REMOVE (R)	the following pages from your manual and REPLACE them with the SAME-NUMBERED pages from this packet wherever you see (R) next to the listed pages.
R	Title page
R	Table of Contents
R	3-4
R	7-16

A vertical revision bar (|) in the outside margin on updated pages shows the latest technical changes made to these pages.

Please file this cover sheet at the back of your document so you will have a record of the changes.

1. About This Manual

1.1 Purpose

This manual describes how to use the CA-MetaCOBOL+ Quality Assurance Facility.

This manual assumes familiarity with the COBOL language and CA-MetaCOBOL+. It provides descriptions, examples, and instructions on how to use the CA-MetaCOBOL+ Quality Assurance Facility.

1.2 Organization

Chapter	**Description**
1	Discusses the purpose of the manual, gives a list of CA-MetaCOBOL+ documentation, and explains notation conventions for CA-MetaCOBOL+.
2	Gives an overview of the Quality Assurance Facility. Describes how to execute this facility and generate an output source program.

Chapter	Description
3	Explains the basic functions, reports, and customization procedures of Quality Assurance for COBOL and DL COBOL.
4	Explains the basic functions, reports, and customization procedures of Quality Assurance for COBOL and DL COBOL when structured programming is used.
5	Explains the basic functions, reports, and customization procedures of Quality Assurance for VS COBOL II and DL VS COBOL II.
Appendix A	Provides an example of the Quality Assurance Facility. Input and output listings are given.
Appendix B	Lists and explains diagnostic messages.
Index	Gives page references for important topics covered in this manual.

1.3 Publications

In addition to this manual, the following publications are supplied with CA-MetaCOBOL+.

Title	Contents
Introduction to CA-MetaCOBOL+	Introduces the CA-MetaCOBOL+ Work Bench, Structured Programming Facility, Quality Assurance Facility, CA-DATACOM/DB Facility, Macro Facility, Panel Definition Facility, and the Online Programming Language.
Installation Guide - MVS	Explains how to install CA-MetaCOBOL+ in the MVS environment.
CA-ACTIVATOR Installation Supplement – MVS	Explains how to install CA-MetaCOBOL+ in the MVS environment using CA-ACTIVATOR.
Installation Guide – VSE	Explains how to install CA-MetaCOBOL+ in the VSE environment.

Title	Contents
Installation Guide – CMS	Explains how to install CA-MetaCOBOL+ in the

	CMS environment.
User Guide	Explains how to customize, get started, and use CA-MetaCOBOL+. Includes information on keyword expansion, the CA-MetaCOBOL+ translator, and CA macro sets and programs.
Structured Programming Guide	Introduces the Structured Programming Facility. Includes information on creating, testing, and maintaining structured programs.
Macro Facility Tutorial	Introduces the Macro Facility. Includes information on writing basic macros, model programming, macro writing techniques, and debugging.
Macro Facility Reference	Includes detailed information on the program flow of the CA-MetaCOBOL+ macro translator, macro format, definition of comments, macro nesting, macro prototypes, symbolic words, and model programming.
Program Development Guide CA-DATACOM/DB	Includes all the information necessary to develop programs that make full use of the functions and features of the CA-DATACOB/DB environment.
Program Development Reference CA-DATACOM/DB	Contains all CA-DATACOM/DB Facility constructs and statements.
Panel Definition Facility Command Reference	Contains all Panel Definition Facility commands.
Panel Definition Facility User Guide	Includes all the information necessary to create, edit, duplicate, rename, delete, index, and print panel definitions and members. Also describes how to generate BMS source.
Online Programming Language Reference	Contains all Online Programming Language statements.
Online Programming Language Guide	Provides further instruction for using Online Programming Language statements.
PC User Guide	Explains how to use CA-MetaCOBOL+/PC. Includes information on the CA-MetaCOBOL+ translator and CA macro sets and programs. Also decribes the relationship between CA-MetaCOBOL+ and CA-MetaCOBOL+/PC.
Program Development Guide CA-DATACOM/PC	Describes how to develop programs that use the CA-DATACOM/PC environment.
String Manipulation Language Guide	Introduces the String Manipulation Language, which provides string handling and inspection

capabilities unavailable in COBOL.

All manuals are updated as required. Instructions accompany each update package.

1.4 Notation Conventions

The following conventions are used in the command formats throughout this manual:

UPPERCASE — is used to display commands or keywords you must code exactly as shown.

lowercase italic — is used to display information you must supply. For example, DASD space parameters may appear as *xxxxxxx xxxxxxx xxxxxxx*.

Underscores — either show a default value in a screen image or represent the highlighting of a word in a screen image.

Brackets [ ] — mean that you can select one of the items enclosed by the brackets; none of the enclosed items is required.

Braces { } — mean that you must select one of the items enclosed by the braces.

Vertical Bar | — separates options. One vertical bar separates two options, two vertical bars separate three options, and so on. You must select one of the options.

Ellipsis . . . — means that you can repeat the word or clause that immediately precedes the ellipsis.

1.5 Summary of Revisions

Minor technical and editorial changes have been made throughout the manual.

2. Overview

CA-MetaCOBOL+'s Quality Assurance (QA) Facility is designed to audit COBOL source for the purpose of improving readability, enforcing organization standards, assuring consistent language usage, and promoting good programming practices. In addition, program documentation can be produced to provide managers and future maintenance programmers with additional information about the program.

The Quality Assurance Facility addresses quality control of standard COBOL source or high-level COBOL source (COBOL containing CA-MetaCOBOL+'s structured verbs and/or data manipulation language). The Quality Assurance macro sets are tailored to installation standards. The QA Facility actively modifies program source when the modification will not alter program logic or operation. Some of the benefits are:

Improved Readability

- Consistent formatting promotes readability and ease of maintenance by the indentation of continued sentences and conditional statements.

- Paragraph names can be prefixed with a sequence number to aid in navigation through the procedural code.

Enforcement of Organization Standards

- The entire program is audited and, where possible, modified to conform to organization standards. When automatic source modification would not provide a clean, maintainable program logic, a diagnostic is issued identifying the violation. Programs not conforming to organization standards can be prevented from entering production.

Control of Language Usage

- COBOL has flexible rules regarding syntax specification. Many keywords have multiple forms (PIC or PICTURE). Data definition clauses can be in any order. Level numbers can be specified inconsistently, etc. The QA Facility modifies the source to conform to the syntax rules specified by your organization.
- Prohibiting some language elements, particularly deviations from ANSI standards (vendor extensions), allows greater portability and easier upgrade to future changes in the language.

Promotion of Good Programming Practices

- Prohibition of language elements that tend to obscure program logic (ALTER, RENAMES, MOVE CORRESPONDING, GO TO DEPENDING) can be desirable. The QA Facility converts ALTER to more maintainable code and flags other language elements.

- Programs can be prepared for evolution to new language standards by identifying elements marked for deletion in proposed new standards and flagging use of those elements today.

Program Documentation

- Data Division mapping provides easy to read, supplemental information about the program to aid in future maintenance.

- A summary of the Quality Assurance and standards audit process is produced, identifying standards violations and language usage information helpful to both programmers and managers.

User Extendability

- The QA Facility is an application of the CA-MetaCOBOL+ Translator. The rules for quality assurance and control are written in the CA-MetaCOBOL+ language and are delivered in source form to allow easy modification or extension. Access to the full range of CA-MetaCOBOL+ macro programming facilities is provided.

2.1 Before You Begin

Before using the QA Facility, please consider the following:

- The QA Facility can be executed to meet one of two objectives: Quality assurance checking only, or quality assurance and standardization:

  - If your objective is only to assure the quality of your programs, there is no need to create a modified source program. By specifying the NODECK translate-time option, the QA Facility only checks the input source. A modified source program is not created.

  - If your objective is to standardize and assure the quality of your programs, you will want to create a modified source program. By specifying the DECK translate-time option, the QA Facility checks and reformats the input source, and creates a modified source program.

- Several options are available to control the processing of COBOL copy books and CA-LIBRARIAN INCLUDE modules. The option chosen depends upon your objective (quality assurance only, or quality assurance and standardization):

- By specifying the COPY=ACTIVE or -INC translate-time option, data definitions are checked and the COPY statements expanded. This option is most effective when executing the QA Facility for quality assurance checking only.

- By specifying the COPY=PASSIVE translate-time option, data definitions are checked, but the COPY statements are not expanded. Although this option can be specified for quality assurance checking only, it is most useful when creating a standardized, translated source program. (CA-LIBRARIAN NO-INC option is the installation default and cannot be specified; it can only be overridden via -INC.)

Note: Translate-time options are described in Section 2.2.2.

2.2 Executing the QA Facility

The CA-MetaCOBOL+ Translator is a general-purpose COBOL macro processor that can be customized to solve a broad range of specific needs. An input job stream is needed to execute the Translator. The input consists of:

- optional translate-time options that control Translator functions for the duration of translation

- optional user-written macro definitions

- QA Facility macro definitions (CQA or SQA) that control the actual input source code translation process

- a COBOL input source program to be translated.

The translate-time options and macro definitions provide specific rules that are used during translation of the input source program. For example, the ALTER verb has been deleted from the COBOL language. Macros provide the rules required to correct source programs that contain ALTER verbs so that the result is an output source program ready for compilation.

Translation begins after the first division header is detected. During translation, each word in the input source program is compared with the rules specified by the translate-time options and macro definitions. Input source words that match are replaced or NOTEd according to the rules; input source words that do not match pass unchanged to the output source program.

All macro definitions and the source program are listed in an Input Listing. Error, warning, and advisory diagnostics are issued as NOTEs following the input record causing the diagnostic. The primary output of the Translator is a modified COBOL output source program, generated in the order of the input source program, and output to an Output Listing.

2.2.1 The JCL

CA-MetaCOBOL+ runs as a batch program. The following sections provide JCL examples for executing CA-MetaCOBOL+ in MVS and VSE environments. Refer to the CA-MetaCOBOL+ *User Guide* for complete details concerning MVS, VSE, and CMS operating considerations.

MVS Environment

The following JCL is provided as a guide for executing CA-MetaCOBOL+ in an MVS environment:

Note: The JCL below is sample start-up JCL only. You must supply the *options*, *macros*, and *Source Program* indicated in the last three lines of the sample below. Refer to the CA-MetaCOBOL+ *User Guide* for more information. If you are using CA-MetaCOBOL+/PC, refer to the CA-MetaCOBOL+/PC *User Guide* for equivalent SET statements.

```
 //[stepname]  EXEC PGM=usermct[,PARM='options']
 //STEPLIB     DD   DSN=user.loadlib,DISP=SHR
 //            DD   DSN=LIBRARIAN.loadlib,DISP=SHR
 //LSTIN       DD   SYSOUT=A[,DCB=BLKSIZE=121]
 //PUNCHF      DD   SYSOUT=B[,DCB=BLKSIZE=80]
 //FE          DD   UNIT=SYSDA,SPACE=(TRK,(20,10))[,DCB=BLKSIZE=148]
 //FM          DD   UNIT=SYSDA,SPACE=(TRK,(20,10))[,DCB=BLKSIZE=148]
[//UNLOAD      DD   DSN=user.maclib,UNIT=SYSDA,SPACE=(TRK,10),]
[//                DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),DISP=(,CATLG,DELETE)]

[//RELOAD      DD   DSN=user.maclib,DISP=SHR]
[//SYSLIB      DD   DSN=user.srclib,DISP=SHR]
 //USERLIB     DD   DSN=user.copylib
[//MASTER      DD   DSN=user.disklibr,DISP=SHR]
 //CARDF       DD   *[,DCB=BLKSIZE=80]
[OPTION options]
[macros]
Source Program
/*
```

You may want to alter the SYSOUT class, BLKSIZE (in multiples of sizes shown), SPACE, and UNIT parameters. The PUNCHF DD statement is not required if the NODECK translate-time option is specified. The use of UNLOAD and RELOAD offer benefits described in the CA-MetaCOBOL+ *User Guide*.

It is recommended that the JCL be placed in a PROC, such as the following:

```
//METACBL  PROC META=MCT,                CA-MetaCOBOL+ Program Name
//              STEPLIB='MCT.LOAD',      Load Library Name
//              COBOL=NULLFILE,          COBOL Output Data Set
//              COPYLIB='MCT.SYSLIB',    COPY Library
//              LIBMSTR='MCT.DISKLIBR',  Disk LIBRARIAN Master
//              TERM=NULLFILE,           Terminal Data Set
//MCT      EXEC PGM=&META
//STEPLIB  DD   DSN=&STEPLIB,DISP=SHR
//LSTIN    DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605)
//PUNCHF   DD   DSN=&COBOL,UNIT=SYSDA,SPACE=(TRK,100),
//              DCB=BLKSIZE=400,DISP=(,PASS)
//SYSLIB   DD   DSN=&COPYLIB,DISP=SHR
//MASTER   DD   DSN=&LIBMSTR,DISP=SHR
//FE       DD   UNIT=SYSDA,SPACE=(TRK,(20,10)),DCB=BLKSIZE=1480
//FM       DD   UNIT=SYSDA,SPACE=(TRK,(20,10)),DCB=BLKSIZE=1480
//RELOAD   DD   DSN=user.maclib,DISP=(OLD,KEEP)
//CARDF    DD   DDNAME=SYSIN
```

VSE Environment

The following JCL is provided as a guide for executing CA-MetaCOBOL+ in a VSE environment:

Note: The JCL below is sample start-up JCL only. You must supply the *options*, *macros*, and *Source Program* indicated in the last three lines of the sample below. Refer to the CA-MetaCOBOL+ *User Guide* for more information. If you are using CA-MetaCOBOL+/PC, refer to the CA-MetaCOBOL+/PC *User Guide* for equivalent SET statements.

```
// LIBDEF ...
// DLBL   IJSYS01,...
// EXTENT SYS001,...                        Work File
// ASSGN  SYS001,DISK,VOL=volserno,SHR
// DLBL   IJSYS02,...
// EXTENT SYS002,...                        Work File
// ASSGN  SYS002,DISK,VOL=volserno,SHR
// DLBL   IJSYS06,...                To 'punch' CA-MetaCOBOL+
// EXTENT SYS006,...                        output to a disk
// ASSGN  SYS006,DISK,VOL=volserno,SHR
// DLBL   IJSYS07,...
// EXTENT SYS007                            Auxiliary output
// ASSGN  SYS007,X'cuu'
// DLBL   IJSYSSL,...                       Private source
// EXTENT SYSSLB,...                 statement library
// ASSGN  SYSSLB,...
// EXEC ZMCTA
     [OPTION options]
     [macros]
     source program
/*
```

Depending on the type of translation, other files may be required (e.g., standard labels for the disk work areas can be used). SYS006 is not required if the NODECK translate-time option is specified.

2.2.2 Translate-time Options

Translate-time options control Translator functions that are to remain in effect for the duration of the translation. Some of these options override installation defaults.

In MVS, translate-time options can be specified in the PARM field of the EXEC statement or on one or more OPTION cards. The options specified on an OPTION card override those specified in the PARM field or preceding OPTION card(s). In VSE, the options must be specified on one or more OPTION cards.

The OPTION card(s) must be the initial input to the Translator following the JCL. The word OPTION must be in columns 1-11, followed by one or more spaces (columns 1-6 can contain a sequence number). Options must be separated by a comma, with no intervening spaces, and no option can extend beyond column 71 or be continued to the next line. More than one OPTION card can be used.

The following list provides the keyword and operands for translate-time options applicable to the QA Facility, along with a brief description of each. Prior to implementing any of these options, refer to the CA-MetaCOBOL+ *User Guide* for a complete description.

Format:

```
{QUOTE}
{APOST}
```

This command specifies the character used, apostrophe or quote, to bind non-numeric literals output by CA-MetaCOBOL+.

Format:

```
{XCOM   }
{ICOM   }
{COMMENT}
```

This command specifies whether or not ANSI COBOL REMARKS paragraphs or NOTEs are converted to comments.

Format:

```
      {IGNORE }
COPY= {ACTIVE }
      {PASSIVE}
```

This command controls the processing of COBOL COPY statements in the input source program. COPY=PASSIVE is recommended for the QA Facility macro sets.

Format:

```
CRSTACK=n
```

This command specifies the amount of space in bytes to be provided at translate-time to stack the words from the REPLACING clause of a COBOL COPY statement.

Format:

```
{NODECK}
{DECK  }
```

This command specifies whether or not the output source program is created.

Format:

```
DEPTH= {nn}
       {66}
```

This command specifies the number of lines per report page.

Format:

```
                {2}
                {3}
      DIALECT=  {4}   {O}
                {V}   {D}
                {X}
                {W}
                {Y}
```

This command specifies which words in the input source program are recognized by the Translator as verbs, words that qualify as keywords for formatting, words that are recognized as special register names, and the syntax checking rules that apply.

Format:

```
               {ADR               }
               {CLOSED            }
      FORMAT=  {name              }
               {(code[,code]...)}
               {OPEN              }
```

This command controls the format of the translated COBOL output source program (Appendix C of the CA-MetaCOBOL+ *User Guide* provides complete details).

Format:

```
          {*ID    }
          {*MC    }
      ID= {*SAVE  }
          {*SEQ   }
          {name   }
          {*BLANK }
```

This command controls the contents of columns 73-80 of the output source records, except CBL statements.

Format:

```
      -INC
```

This command activates the processing of LIBRARIAN -INC statements in an input source program retrieved with a *$LIBED translator-directing statement (refer to Section 2.3).

Format:

```
      INVDEC
```

This command specifies that CA-MetaCOBOL+ recognizes a comma as the decimal character.

Format:

```
      LSEQ
```

This command causes the contents of the input source sequence field to appear to the right of the output source in the Output Listing.

Format:

```
              {NUM}
      RESEQ= {n  }
              {0  }
```

This command controls the sequence numbering of columns 1-6 in the output source program.

Format:

```
 {SEPPAR  }
 {NOSEPPAR}
```

This command separates subscripts from data-names through the insertion of a space character.

2.2.3 User-written Macro Definitions

Macro definitions control the actual source code translation process. User-written macro definitions must follow translate-time options, if specified, and precede the QA Facility macro set in the input. They can be coded directly or obtained from another source through the use of translator-directing statements (refer to Section 2.3). Refer to Section 2.5 and the *Macro Facility Reference* for more information.

2.2.4 QA Facility Macro Definitions

The QA Facility macro sets contain modifiable macro definitions that perform functions specific to the COBOL dialect. They must follow translate-time options and user-written macro definitions, if specified, and precede the input source program in the input. They can be included directly in the input or obtained from another source through the use of translator-directing statements (refer to Section 2.3).

For more information concerning functions and modification of the QA Facility macro sets, refer to Sections 2.5, Chapter 3, and Chapter 4.

2.2.5 The Input Source Program

The input source program is the last item input to the Translator. It must be a valid COBOL program that will compile without errors and must begin with IDENTIFICATION DIVISION or ID DIVISION. The input source program must follow macro definitions in the input; it can be coded directly or obtained from another source through the use of translator-directing statements (refer to Section 2.3).

2.3 Translator-Directing Statements

Translator-directing statements applicable to QA Facility are used to retrieve user-written macro definitions, if specified, QA Facility macro sets, and/or an input source program. Translator-directing statements are embedded as special COBOL comment records containing an asterisk (*) in the continuation column (column 7), followed by a dollar sign ($). They must follow the OPTION cards and precede the first COBOL division header.
Translator-directing statements applicable to QA Facility are listed as follows, along with a brief description of each. Prior to using any of these statements, refer to the CA-MetaCOBOL+ *User Guide* for a complete description.

STATEMENT	**DESCRIPTION**
*$COPY *module-name* [*commentary*]	Retrieves input from SYSLIB in MVS, or from the source library in VSE.
*$LIBED *module-name*	Retrieves input from CA-LIBRARIAN master disk or CA-PANVALET.
*$LIBET *module-name*	Retrieves input from CA-LIBRARIAN master tape.

2.4 Generating an Output Source Program

The COBOL output source program is generated only when the DECK translate-time option is specified (refer to Section 2.2.2). In MVS, the output source program is written to the PUNCHF data set; in VSE, SYS006. Other translate-time options control formatting, COBOL sequencing, and the contents of columns 73-80.

2.5 Extending the QA Facility

QA Facility macro sets are written in CA-MetaCOBOL+ macro programming language and distributed in source form. There may be instances where QA Facility does not meet certain requirements specified by your site. By modifying macros contained in the macro set, you can extend QA Facility to accommodate these needs. In addition, you can create your own macros to be used in conjunction with a QA Facility macro set (in this case, the macros must precede the macro set in the input).

A complete understanding of macro writing is strongly recommended prior to creating and modifying macros contained in a macro set. The *Macro Facility Reference* and *Macro Facility Tutorial* provide complete details.

Format:

W [*c*] *macro-name* : [*model-word*] . . .

W

Specifies the macro type code for *word.* W must appear in the continuation column (column 7).

c

Represents the division code, which can be one or any combination of I, E, D, or P, or blank specifying all divisions.

macro-name

Represents any string of 30 or less characters which is not a literal or a single special character.

:

The required separator.

model-word

Represents the rules to be acted upon whenever a call is made to the macro.

For example, word macros can be used to make cryptic names more meaningful:

```
      WP  CHW : COMPUTE-HOURLY-WAGES.
```

In this example, the word CHW is searched for only in the PROCEDURE DIVISION of the input source program. For each occurrence, the macro is called and the word COMPUTE-HOURLY-WAGES. is output in its place.

Note that even if the word CHW in the input source program is followed by a period, only one terminating period will appear in the output source program. The Translator suppresses consecutive terminating periods.

2.6 Sample Input to the Translator

The following example illustrates sample input to the CA-MetaCOBOL+ Translator in a CMS, MVS, VSE, PC-DOS, or MS-DOS environment:

```
OPTION ID=*BLANK
    W   X1 : INDEX-1
    W   X2 : INDEX-2
    W   X3 : INDEX-3
    W   A-YEAR : ACCEPT-YEAR
    W   A-MONTH : ACCEPT-MONTH
    W   A-DAY : ACCEPT-DAY
    *$COPY CQA
    *$COPY SAMPPGM
```

The OPTION card is followed by word macros.

3. COBOL Quality Assurance (CQA) and DL COBOL Quality Assurance (DCQA)

COBOL Quality Assurance (CQA) provides functions for standardizing COBOL source programs. DL COBOL Quality Assurance (DCQA) provides functions for standardizing COBOL source programs that contain DL verbs. Unless noted, the term CQA refers to both COBOL Quality Assurance and DL COBOL Quality Assurance.

The following sections contain descriptions of all functions performed by CQA and the methods available for customizing CQA to your needs.

3.1 Functions

Functions performed by CQA have either an *active* or *passive* effect on a COBOL program. Active functions automatically change the source code. Passive functions examine source code and provide you with a NOTE describing conditions that you may wish to change manually or with a macro.

NOTE diagnostics are listed in the Input or Output Listing in the form:

```
*******  NOTE   N99  CQAnnz message
```

CQA represents the CQA diagnostic abbreviation

nn represents the diagnostic number unique within CQA

z represents the diagnostic severity code, and

message represents a brief description of the diagnostic.

A complete explanation of NOTE diagnostics can be found in Appendix B.

The remainder of this section lists each COBOL element examined and the action taken or condition NOTEd by CQA. The default action or issuance of a NOTE can be overridden via the control statement referenced or changed by removing the diagnostic within the macro set.

Note: All control statements begin with $CQA-. Control statements and diagnostic removal are described in Section 3.3.

Listed below are the source elements and the functions they perform.

General

Data Definition Clause Order

Data definition clauses are always arranged in the sequence shown below. Note that RENAMES and REDEFINES are mutually exclusive and always placed first when coded; the other clauses do not have a pre-defined COBOL precedence. ASCENDING/DESENDING KEY clauses are part of the OCCURS clause.

level number
data-name
RENAMES
REDEFINES
OCCURS
PICTURE
SIGN
BLANK WHEN ZERO
JUSTIFIED
USAGE
SYNCHRONIZED
VALUE

IBM Extensions

IBM extensions to the 1974 ANSI COBOL standard are NOTEd. See $CQA-EXTEN.

Environment Division

SAME AREA Clause

Two or more data sets allocated to the same I/O area via the SAME AREA clause are NOTEd; they impose restrictions on block size, record format, and program logic.

Data Division

General

Data Division file and record descriptions are *mapped* in the Auxiliary Listing. See $CQA-MAP.

File Descriptions

BLOCK CONTAINS clauses specified for sequential files that do not contain a value of 0 and/or RECORD CONTAINS clauses containing a value of 0 are NOTEd.

Level Numbers

Data Definition level numbers are made consistent relative to the data hierarchy. See $CQA-INCR in Section 3.3.2.

Data-Names

Data-names shorter than the length specified by $CQA-SHORT are NOTEd. See $CQA-SHORT in Section 3.3.2.

Reserved Words

Abbreviations or full spellings are generated (e.g., PIC, COMP or PICTURE, COMPUTATIONAL). See $CQA-ABBR in Section 3.3.2.

Optional words IS, USAGE, BY, ON, TIMES, KEY, CHARACTER, WHEN, and SIGN are added to data attribute clauses (e.g., PICTURE IS, USAGE IS, etc.) or deleted. See $CQA-WORD in Section 3.3.2.

Group Items

Group item USAGE and SYNCHRONIZED clauses are moved to their subordinate elementary items.

Group FILLER Items

Group FILLER items referenced in the data hierarchy are NOTEd.

OCCURS...DEPENDING ON Clause

OCCURS...DEPENDING ON clauses within a record description can be confusing to the maintenance programmer. Nested OCCURS...DEPENDING ON clauses are NOTEd separately.

PICTURE and VALUE Clause

All numeric items that are COMPUTATIONAL, COMPUTATIONAL-3, or COMPUTATIONAL-4 are signed so the compiler does not generate additional instructions to remove and restore the signs. See $CQA-SIGN in Section 3.3.2.

All COMPUTATIONAL or COMPUTATIONAL-4 binary data items are changed to the number of digits necessary to fill the full amount of storage allocated to them: 4, 9, or 18. All COMPUTATIONAL-3 packed data items with an even number of digits in the picture, except those with a length of 18, are increased to an odd number of digits. See $CQA-SIZE in Section 3.3.2.

REDEFINES Clause

The REDEFINES object is made to conform to 1974 ANSI COBOL standards. If this action is overridden, non-conformance is NOTEd. See $CQA-REDEF in Section 3.3.2.

RENAMES Entry

RENAMES entries are NOTEd.

SYNCHRONIZED Clause

All binary, index, and floating point data items are SYNCHRONIZED to avoid the generation of alignment logic. If this action is overridden, non-synchronization is NOTEd. See $CQA-SYNC in Section 3.3.2.

VALUE Clause

VALUE IS SPACE or VALUE IS ZERO is added for Working-Storage items lacking a value other than those subordinate to a REDEFINES or OCCURS clause.

Procedure Division

Procedure-Names

Procedure-names shorter than the length specified by $CQA-SHORT are NOTEd as potentially non-meaningful. See $CQA-SHORT in Section 3.3.2.

Sequentially numbered prefixes are assigned to all procedure-name definitions and all references to them are altered. For example, the statement PERFORM MINOR-BREAK in the paragraph CHECK-FOR-BREAK becomes PERFORM 0400-MINOR-BREAK in the paragraph 0200-CHECK-FOR-BREAK. Instead of a cross-reference listing, the sequential prefix can then be used to locate the new logic path.

The following standards are used in numbering procedure-names:

- The resulting procedure-name will be a 4-digit number, a hyphen, and the first 25 characters of the original procedure-name. If the original name is actually longer than 25 characters and truncation causes the name to end with a hyphen, the character X is substituted for the hyphen.

- Procedure-names that are already numbered (1 to 4 digits followed by a hyphen) are re-sequenced by replacing the old number with a new number.

- Numeric procedure-names are suffixed with -PARA and then numbered; the original number is retained.

The increment for the paragraph sequence number is specified in $CQA-SEQN.

ACCEPT Verb

ACCEPT FROM CONSOLE is NOTEd as inefficient and prone to operator error.

ALTER Verb

A switch is added in Working-Storage for each ALTERed paragraph, and each ALTER statement is replaced with a MOVE statement that sets the switch. An IF statement testing the switch is then added prior to the GO TO statement in the ALTERed paragraph.

Arithmetic Statements

DISPLAY items and operands of inconsistent numeric usages and/or decimal alignments are NOTEd when encountered in arithmetic statements.

Conditional Statements

Group items used in an IF statement and data items of differing lengths that require padding before comparison are NOTEd.

CORRESPONDING Option

CORRESPONDING in an ADD, SUBTRACT, or MOVE statement is NOTEd as a potential maintenance problem.

Debugging Verbs

READY TRACE, RESET TRACE, EXHIBIT, and ON statements within production programs are NOTEd.

DISPLAY Verb

DISPLAY UPON CONSOLE is NOTEd as inefficient and prone to operator error.

GO TO Statements

A programmer written or CQA-generated (in ALTER conversion) GO TO statement and its forward or backward direction are NOTEd.

INSPECT, EXAMINE, and TRANSFORM Verbs

Use of execution resources by these verbs is costly and therefore NOTEd.

I/O Statements

The use and/or failure to use FROM and INTO options is NOTEd.

ON SIZE ERROR Clause

NOTEd as inefficient and an indication of inadequate data editing.

PERFORM Verb and SORT Procedure Invocation

PERFORM verbs and SORT procedure invocations specified with or without the THRU option are NOTEd. The direction of the routines is NOTEd as either backward or forward in the program.

Program Segmentation

Overlay structures are NOTEd as having questionable value in a virtual system.

STOP Literal

NOTEd as prone to operator error and invalidates the use of the program as a sub-program.

Subscripts/Indices

Subscripts used to accomplish sequential searching of a table are NOTEd; sequential searching of a table should be accomplished with a SEARCH statement. Subscripts defined as COMP-3 or DISPLAY data items are NOTEd as inefficient.

3.2 Reports

In addition to the Input and Output Listings generated by the Translator, CQA generates an Auxiliary Listing that contains a Summary Report and, if specified, a Map Report.

3.2.1 Summary Report

CQA automatically generates a Summary Report in the Auxiliary Listing. The CQA Summary Report provides a summary of all macro-generated diagnostics, as shown in the following example:

```
-------------------------------------------------------------
|** COBOL QUALITY ASSURANCE DIAGNOSTIC SUMMARY               |
|**                                                         |
|** CQA39-'GO TO' VIOLATIONS                              10|
|** CQA40-FORWARD BRANCHES (GO TO)                         7|
|** CQA41-BACKWARD BRANCHES (GO TO)                        8|
|** CQA44-'PERFORM' OR 'SORT PROCEDURE' WITHOUT 'THRU'     6|
|** CQA45-'PERFORM' OR 'SORT PROCEDURE' WITH 'THRU'        7|
|** CQA46-FORWARD 'PERFORM' OR 'SORT PROCEDURE'            6|
|** CQA47-BACKWARD 'PERFORM' OR 'SORT PROCEDURE'           7|
|** CQA60A-SHORT DATA-NAMES OR PROCEDURE-NAMES             3|
|** CQA61-DATA-NAME TRUNCATIONS                            3|
|** CQA63-UNDEFINED PROCEDURE-NAMES                        1|
-------------------------------------------------------------
```

Diagnostics can be modified or removed from the CQA macro set for specific programming needs. Refer to Section 3.3.1 for more information.

3.2.2 Map Report

Data areas to be mapped are specified by the $CQA-MAP control statement (refer to Section 3.3.2). An example is shown below.

```
LEVEL  NAME               USAGE  BYTES  POSN OCCUR OCCLV REDEF  VALUE  SIGN    SYNC  DIGIT  DECML

       WORKING-STORAGE
01     WORK-AREA           GRP   80     1                       VALUE
  02   TRANSACTION-CODE    A/N   1      11                      VALUE
  02   PLANT               NUM   3      20                      VALUE             3         0
  02   PRODUCT-CODE        A/N   1      35                      VALUE
  02   MAKE-SHIP-CODE      A/N   1      56                      VALUE
  02   GROSS               NUM   5      66                      VALUE             5         0
01     COUNTERS            GRP   30     1                       VALUE
  02   PLANT-ADDITIONS     PACK  3      1                       VALUE
  02   PLANT-CHANGES       PACK  3      4                       VALUE  SIGN       5         0
  02   PLANT-DELETES       PACK  3      7                       VALUE  SIGN       5         0
  02   PLANT-GROSS-MADE    PACK  3      10                      VALUE  SIGN       5         0
  02   PLANT-GROSS-SHIP    PACK  3      13                      VALUE  SIGN       5         0
  02   FINAL-ADDITIONS     PACK  3      16                      VALUE  SIGN       5         0
  02   FINAL-CHANGES       PACK  3      19                      VALUE  SIGN       5         0
  02   FINAL-DELETES       PACK  3      22                      VALUE  SIGN       5         0
  02   FINAL-GROSS-MADE    PACK  3      25                      VALUE  SIGN       5         0
  02   FINAL-GROSS-SHIP    PACK  3      28                      VALUE  SIGN       5         0
01     REPORT-RECORD       GRP   120    1                       VALUE
  02   REPORT-PLANT        A/N     3    6                       VALUE
  02   REPORT-ADDITIONS    NUM     5    14                      VALUE             5         0
  02   REPORT-DELETES      NUM     5    34                      VALUE             5         0
  02   REPORT-GROSS-SHIP   NUM     5    54                      VALUE             5         0
  02   REPORT-FINAL        A/N     5    64                      VALUE
```

LEVEL is the level number or character that defines group items, hierarchy, and special entries in the Data Division.

NAME is the section-name, data-name, and names of subordinate data items.

USAGE is the form in which the data is stored.

BYTES is the length.

POSN is the position within the record. The first position is 1.

OCCUR is the number of specified occurrences.

OCCLV is the OCCURS nesting level.

REDEF clause is indicated.

VALUE clause is indicated.

SIGN clause is indicated.

SYNC clause is indicated.

DIGIT the number of decimal digits represented by numeric items.

DECML decimal places represented by numeric items.

3.3 Customization

The functions performed by CQA are provided as default values in the CQA macro set. Prior to executing CQA, you must determine which functions do not apply to your specific programming situations and/or local standards, and then tailor CQA accordingly. There are two methods available for customizing CQA:

- The macro set was designed to anticipate all programming options that a programmer may or may not choose to implement. For this reason, there are cases where CQA will issue two or more contradictory NOTE diagnostics. The macro set can be modified by removing diagnostics that do not apply.

- CQA provides programming alternatives to override some of the macro set default values. These alternatives must be determined and then implemented through the specification of control statements.

3.3.1 Macro Set Diagnostics

Contradictory diagnostics must be removed from the macro set. For example, in the case of the PERFORM statement, the following will be generated in the Input Listing for each applicable occurrence:

```
******* NOTE   N99   CQA44A 'PERFORM' WITHOUT THRU PHRASE
******* NOTE   N99   CQA45A 'PERFORM' WITH THRU PHRASE
```

All CQA diagnostics are identified with the diagnostic number in columns 73-77 of the macro set. Any diagnostic can be removed by locating the diagnostic number, and either deleting the lines of code or placing an asterisk in column 7 to comment them out; it is recommended that diagnostics be commented out. There will always be two or more lines of code, and in some cases they can be widely separated. Note that deleting the lines only removes the diagnostic: it does not delete the functional code.

In addition to CQA44A/CQA45A, other contradictory diagnostics are CQA14W/CQA15W and CQA48A/CQA49A. Locate the diagnostic numbers in columns 73-77 of the macro set, and select and remove the appropriate diagnostic.

3.3.2 Control Statements

Control statements override some macro set default values. They are coded as execution defaults in Area B (columns 12-72) following the macro set.

They can also be embedded in the Identification Division.

Control statements are divided into two groups: those that specify coding standards and those that provide column formatting.

Coding Standards

CQA functions that affect COBOL coding standards ensure that the code is reasonably efficient and that it does not contain obsolete language elements. Where indicated, macro set default values for these types of functions are shown as underlined parameters of the following control statements, and can be overridden by specifying the alternative.

Format:

```
$CQA-ABBR {Y}
          {N}
```

Specifies keyword abbreviations or full spellings:

Y

PIC, COMP, COMP-n, SYNC, and JUST.

N

PICTURE, COMPUTATIONAL, COMPUTATIONAL-n, SYNCHRONIZED, and JUSTIFIED.

Format:

```
              {YES}
$CQA-DLVERBS  {Y  }
              {NO }
              {N  }
```

This option is only available under DCQA. It specifies how verbs are standardized:

Y

or YES means that verbs are standardized as DL verbs.

N

or NO means that verbs are standardized as COBOL verbs.

Format:

```
$CQA-EXTEN {Y}
           {N}
```

Specifies whether IBM extension are NOTEd or ignored:

Y

IBM extensions retained for compatibility with the SYSTEM/360 compilers are NOTEd.

N

No diagnostics are produced.

Format:

```
                 {YES}
$CQA-FROM/INTO {Y  }
                 {NO }
                 {N  }
```

Specifies whether FROM/INTO statements are NOTEd or ignored:

Y

FROM/INTO statements are NOTEd. Y is an abbreviation for YES.

N

No diagnostics are produced. N is an abbreviation for NO.

Format:

```
$CQA-INCR {n       }
          {2[ ODD]}
```

Specifies the level numbering increment to be used in the Data Division:

n

Can be specified as any number from 0 to 10 inclusive. If 0 is specified, level numbers are not changed.

2

Level numbers are incremented as 01, 02, 04, 06, 08, and so on.

2 ODD

Level numbers are incremented as 01, 03, 05, 07, 09, and so on.

Format:

```
         {data-name    [ [data-name    ]       ] }
$CQA-MAP {section-name[ [section-name] . . . ] }
         {DATA         [ [             ]       ] }
```

Defines the data areas to be mapped and listed in the Auxiliary Listing. A maximum of ten areas can be mapped in one translation (see Section 3.2.2). The Report Section and items defined within it cannot be mapped.

data-name

Data-names, all data items subordinate to them, and their associated attributes are listed.

section-name

Section-names, all data items subordinate to them, and their associated attributes are listed, where *section-name* represents FILE, WORKING-STORAGE, LINKAGE, or COMMUNICATION.

DATA

A map of the entire Data Division, excluding the Report Section, is created.

Format:

```
$CQA-REDEF {N}
           {Y}
```

Specifies whether or not correction of the REDEFINES clause to ANSI specifications is to be made:

Y

The object of a REDEFINES clause is corrected.

N

No correction is made.

Format:

```
              {SHORT}
$CQA-RELATIONS {  S  }
              {LONG }
              {  L  }
```

Enables you to set standards for relational operators in conditional statements:

S

and SHORT are synonyms. The long forms of relational operators are converted to short forms. For example, EQUAL TO becomes =, and GREATER THAN becomes >.

L

and LONG are synonyms. The short forms of relational operators are converted to long forms. For example, = becomes EQUAL TO, and > becomes GREATER THAN.

Format:

```
$CQA-SEQN {nnnn}
          {10  }
```

Defines the increment for Procedure Division procedure-name numbering:

nn

Must be between 0 and 1000 inclusive. If 0 is specified, procedure-names are not changed.

Format:

```
$CQA-SHORT {nn}
           {6 }
```

Defines the minimum length of an acceptable data-name or procedure-name:

nn

Must be between 1 and 30 inclusive.

Format:

```
$CQA-SIGN {Y}
          {N}
```

Specifies whether or not all COMPUTATIONAL, COMPUTATIONAL-3, and COMPUTATIONAL-4 items are to be signed:

Y

Signs are added where missing from PICTURE and VALUE clauses.

N

PICTURE and VALUE clauses are not changed.

Format:

```
$CQA-SIZE {Y}
          {N}
```

Specifies numeric item size optimization:

Y

COMPUTATIONAL-3 data items are expanded to an odd number of digits except those with a length of 18. COMPUTATIONAL and COMPUTATIONAL-4 data items are expanded to the number of digits necessary to fill the full amount of storage allocated to them: 4, 9, or 18.

N

Numeric item sizes are not changed.

Format:

```
$CQA-SYNC {Y}
          {N}
```

Specifies the status of data item synchronization:

Y

A SYNCHRONIZED clause is appended to all elementary binary data item descriptions.

N

A warning is issued for each unsynchronized item if a dialect has been specified for any ANSI COBOL compiler earlier than the 1974 standard.

Format:

```
          {YES}
$CQA-THRU {Y  }
          {NO }
          {N  }
```

Specifies whether THRU statements are NOTEd or ignored:

Y

THRU statements are NOTEd. Y is an abbreviation for YES.

N

No diagnostics are produced. N is an abbreviation for NO.

Format:

```
           {YES}
$CQA-VALUE {Y  }
           {NO }
           {N  }
```

Specifies whether values are assigned in the WORKING STORAGE Section:

Y

is an abbreviation for YES. Values are assigned in the WORKING STORAGE Section.

N

is an abbreviation for NO. Values are not assigned in the WORKING STORAGE Section.

Format:

```
$CQA-WORD {Y}
          {N}
```

Specifies whether or not optional keywords IS, USAGE, BY, ON, TIMES, KEY, CHARACTER, WHEN, and SIGN are to be included in the output text:

Y

Words are included.

N

Words are not included.

Format:

```
$CQA-77 wY}
        {N}
```

Specifies whether or not level 77 data items are converted to level 01 items:

Y All level 77 data items are converted.

N No conversion is made.

Formatting Columns

CQA default values for formatting the columns in the output source program can be overridden with the following control statements. For each control statement, *integer* represents the column where the clause is to be aligned, and must be a number from 12 to 72 inclusive. Otherwise, the default is 0, indicating that no columnar alignment is to be executed.

Note: If the first word of the associated clause is placed so that it would extend beyond column 72, it will be right-aligned from column 72. By default, the clause is not aligned to a specific column.

Control Statement	**Assigns Columnar Alignment to:**
$CQA-BLANK-COLUMN *integer*	BLANK clause
$CQA-DEPENDING-COLUMN *integer*	DEPENDING clause
$CQA-INDEXED-COLUMN *integer*	INDEXED clause
$CQA-JUSTIFIED-COLUMN *integer*	JUSTIFIED clause
$CQA-KEY-COLUMN *integer*	ASCENDING/DESCENDING KEY clause
$CQA-OCCURS-COLUMN *integer*	OCCURS clause
$CQA-PICTURE-COLUMN *integer*	PICTURE clause
$CQA-REDEFINES-COLUMN *integer*	REDEFINES clause and RENAMES clause
$CQA-SIGN-COLUMN *integer*	SIGN clause
$CQA-SYNC-COLUMN *integer*	SYNCHRONIZED clause
$CQA-USAGE-COLUMN *integer*	USAGE clause
$CQA-VALUE-COLUMN *integer*	VALUE clause
$CQA-DEFAULT-COLUMN *integer*	All REDEFINES/RENAMES, OCCURS, DEPENDING, KEY, INDEXED, PICTURE, SIGN, BLANK, JUSTIFIED, USAGE, SYNCHRONIZED, and VALUE clauses that have not previously been assigned alignment by the specific control statements defined in this section.

4. Structured Programming Quality Assurance (SQA) and DL Structured Programming Quality Assurance (DSQA)

Structured Programming Quality Assurance (SQA) provides functions for standardizing Structured Programming (SP) Facility source programs. DL Structured Programming Quality Assurance (DSQA) provides functions for standardizing Structured Programming (SP) Facility source programs that contain DL verbs. Unless noted, the term SQA refers to both Structured Programming Quality Assurance and DL Structured Programming Quality Assurance.

The following sections contain descriptions of all functions performed by SQA and the methods available for customizing SQA to your needs.

4.1 Functions

Functions performed by SQA have either an *active* or *passive* effect on a SP Facility program. Active functions automatically change the source code. Passive functions examine source code and provide you with a NOTE describing conditions that you may wish to change manually or with a macro.

NOTE diagnostics are listed in the Input or Output Listing in the form:

******* NOTE N99 SQA*nnz message*

SQA represents the SQA diagnostic abbreviation

nn represents the diagnostic number unique within SQA

z represents the diagnostic severity code

message represents a brief description of the diagnostic.

A complete explanation of NOTE diagnostics can be found in Appendix B.

The remainder of this section lists each SP Facility element examined and the action taken or condition NOTEd by SQA. The default action or issuance of a NOTE can be overridden via the control statement referenced or changed by removing the diagnostic within the macro set.

Note: All control statements begin with $SQA-. Control statements and diagnostic removal are described in Section 4.3

4.1.1 Source Elements and Functions Performed

This section contains a list of each source element in each COBOL Division. Each source element is bolded, and the function it performs is listed directly below the element.

General

Data Definition Clause Order

Data definition clauses are always arranged in the sequence shown below. Note that RENAMES and REDEFINES are mutually exclusive and always placed first when coded; the other clauses do not have a pre-defined COBOL precedence. ASCENDING/DESENDING KEY clauses are part of the OCCURS clause.

level number
data-name
RENAMES
REDEFINES
OCCURS
PICTURE
SIGN
BLANK WHEN ZERO
JUSTIFIED
USAGE
SYNCHRONIZED
VALUE

IBM Extensions

IBM extensions to the 1974 ANSI COBOL standard are NOTEd. See $SQA-EXTEN.

Environment Division

SAME AREA Clause

Two or more data sets allocated to the same I/O area via the SAME AREA clause are NOTEd; they impose restrictions on block size, record format, and program logic.

Data Division

General

Data Division file and record descriptions are *mapped* in the Auxiliary Listing. See $SQA-MAP in Section 4.3.2.

File Descriptions

BLOCK CONTAINS clauses specified for sequential files that do not contain a value of 0 and/or RECORD CONTAINS clauses containing a value of 0 are NOTEd.

Level Numbers

Data Definition level numbers are made consistent relative to the data hierarchy. See $SQA-INCR in Section 4.3.2.

Data-Names

Data-names shorter than the length specified by $SQA-SHORT are NOTEd. See $SQA-SHORT in Section 4.3.2.

Reserved Words

Abbreviations or full spellings are generated (e.g., PIC, COMP or PICTURE, COMPUTATIONAL). See $SQA-ABBR in Section 4.3.2.

Optional words IS, USAGE, BY, ON, TIMES, KEY, CHARACTER, WHEN, and SIGN are added to data attribute clauses (e.g., PICTURE IS, USAGE IS, etc.) or deleted. See $SQA-WORD in Section 4.3.2.

Group Items

Group item USAGE and SYNCHRONIZED clauses are moved to their subordinate elementary items.

Group FILLER Items

Group FILLER items referenced in the data hierarchy are NOTEd.

OCCURS...DEPENDING ON Clause

OCCURS...DEPENDING ON clauses within a record description can be confusing to the maintenance programmer. Nested OCCURS...DEPENDING ON clauses are NOTEd separately.

PICTURE and VALUE Clause

All numeric items that are COMPUTATIONAL, COMPUTATIONAL-3, or COMPUTATIONAL-4 are signed so the compiler does not generate additional instructions to remove and restore the signs. See $SQA-SIGN in Section 4.3.2.

All COMPUTATIONAL or COMPUTATIONAL-4 binary data items are changed to the number of digits necessary to fill the full amount of storage allocated to them: 4, 9, or 18. All COMPUTATIONAL-3 packed data items with an even number of digits in the picture, except those with a length of 18, are increased to an odd number of digits. See $SQA-SIZE.

REDEFINES Clause

The REDEFINES object is made to conform to 1974 ANSI COBOL standards. If this action is overridden, non-conformance is NOTEd. See $SQA-REDEF in Section 4.3.2.

RENAMES Entry

RENAMES entries are NOTEd.

SYNCHRONIZED Clause

All binary, index, and floating point data items are SYNCHRONIZED to avoid the generation of alignment logic. If this action is overridden, non-synchronization is NOTEd. See $SQA-SYNC in Section 4.3.2.

VALUE Clause

VALUE IS SPACE or VALUE IS ZERO is added for Working-Storage items lacking a value other than those subordinate to a REDEFINES or OCCURS clause.

Procedure Division

General

START and END DATA file and record descriptions are mapped in the Auxiliary Listing. See $SQA-MAP in Section 4.3.2.

Procedure-Names

Procedure-names shorter than the length specified by $SQA-SHORT are NOTEd as potentially non-meaningful. See $SQA-SHORT in Section 4.3.2.

Sequentially numbered prefixes are assigned to all procedure-name definitions and all references to them are altered. For example, the statement PERFORM MINOR-BREAK in the paragraph CHECK-FOR-BREAK becomes PERFORM 0400-MINOR-BREAK in the paragraph 0200-CHECK-FOR-BREAK. Instead of a cross-reference listing, the sequential prefix can then be used to locate the new logic path.

The following standards are used in numbering procedure-names:

- The resulting procedure-name will be a 4-digit number, a hyphen, and the first 25 characters of the original procedure-name. If the original name is actually longer than 25 characters and truncation causes the name to end with a hyphen, the character X is substituted for the hyphen.
- Procedure-names that are already numbered (1 to 4 digits followed by a hyphen) are re-sequenced by replacing the old number with a new number.
- Numeric procedure-names are suffixed with -PARA and then numbered; the original number is retained.

The increment for the paragraph sequence number is specified in $SQA-SEQN.

ACCEPT Verb

ACCEPT FROM CONSOLE is NOTEd as inefficient and prone to operator error.

ALTER Verb

A switch is added in Working-Storage for each ALTERed paragraph, and each ALTER statement is replaced with a MOVE statement that sets the switch. An IF statement testing the switch is then added prior to the GO TO statement in the ALTERed paragraph.

Arithmetic Statements

DISPLAY items and operands of inconsistent numeric usages and/or decimal alignments are NOTEd when encountered in arithmetic statements.

Conditional Statements

Group items used in an IF statement and data items of differing lengths that require padding before comparison are NOTEd.

CORRESPONDING Option

CORRESPONDING in an ADD, SUBTRACT, or MOVE statement is NOTEd as a potential maintenance problem.

Debugging Verbs

READY TRACE, RESET TRACE, EXHIBIT, and ON statements within production programs are NOTEd.

DISPLAY Verb

DISPLAY UPON CONSOLE is NOTEd as inefficient and prone to operator error.

INSPECT, EXAMINE, and TRANSFORM Verbs

Use of execution resources by these verbs is costly and therefore NOTEd.

I/O Statements

The use and/or failure to use FROM and INTO options is NOTEd.

PERFORM Verb and SORT Procedure Invocation

PERFORM verbs and SORT procedure invocations specified with or without the THRU option are NOTEd. The direction of the routines is NOTEd as either backward or forward in the program.

Program Segmentation

Overlay structures are NOTEd as having questionable value in a virtual system.

STOP Literal

NOTEd as prone to operator error and invalidates the use of the program as a sub-program.

Subscripts/Indices

Subscripts used to accomplish sequential searching of a table are NOTEd; sequential searching of a table should be accomplished with a SEARCH statement. Subscripts defined as COMP-3 or DISPLAY data items are NOTEd as inefficient.

4.2 Reports

In addition to the Input and Output Listings generated by the Translator, SQA generates an Auxiliary Listing that contains a Summary Report and, if specified, a Map Report.

4.2.1 Summary Report

SQA automatically generates a Summary Report in the Auxiliary Listing. The Summary Report provides a summary of all macro-generated diagnostics, as shown in the following example:

```
---------------------------------------------------------
|** SP QUALITY ASSURANCE DIAGNOSTIC SUMMARY              |
|**                                                      |
|** SQA39-'GO TO' VIOLATIONS                           10|
|** SQA40-FORWARD BRANCHES (GO TO)                      7|
|** SQA41-BACKWARD BRANCHES (GO TO)                     8|
|** SQA44-'PERFORM' OR 'SORT PROCEDURE' WITHOUT 'THRU'  6|
|** SQA45-'PERFORM' OR 'SORT PROCEDURE' WITH 'THRU'     7|
|** SQA46-FORWARD 'PERFORM' OR 'SORT PROCEDURE'         6|
|** SQA47-BACKWARD 'PERFORM' OR 'SORT PROCEDURE'        7|
|** SQA60A-SHORT DATA-NAMES OR PROCEDURE-NAMES          3|
|** SQA61-DATA-NAME TRUNCATIONS                         3|
|** SQA63-UNDEFINED PROCEDURE-NAMES                     1|
---------------------------------------------------------
```

Diagnostics can be modified or removed from the SQA macro set for specific programming needs. Refer to Section 4.3.1 for more information.

4.2.2 Map Report

Data areas to be mapped are specified by the $SQA-MAP control statement (refer to Section 4.3.2). Here is an example:

```
LEVEL  NAME                   USAGE  BYTES  POSN OCCUR OCCLV REDEF VALUE SIGN   SYNC   DIGIT   DECML

         WORKING-STORAGE
01       WORK-AREA              GRP    80      1                  VALUE
  02     TRANSACTION-CODE       A/N    1       11                 VALUE
  02     PLANT                  NUM    3       20                 VALUE            3          0
  02     PRODUCT-CODE           A/N    1       35                 VALUE
  02     MAKE-SHIP-CODE         A/N    1       56                 VALUE
  02     GROSS                  NUM    5       66                 VALUE            5          0
01       COUNTERS               GRP    30      1                  VALUE
  02     PLANT-ADDITIONS        PACK   3       1                  VALUE
  02     PLANT-CHANGES          PACK   3       4                  VALUE SIGN           5         0
  02     PLANT-DELETES          PACK   3       7                  VALUE SIGN           5         0
  02     PLANT-GROSS-MADE       PACK   3       10                 VALUE SIGN           5         0
  02     PLANT-GROSS-SHIP       PACK   3       13                 VALUE SIGN           5         0
  02     FINAL-ADDITIONS        PACK   3       16                 VALUE SIGN           5         0
  02     FINAL-CHANGES          PACK   3       19                 VALUE SIGN           5         0
  02     FINAL-DELETES          PACK   3       22                 VALUE SIGN           5         0
  02     FINAL-GROSS-MADE       PACK   3       25                 VALUE SIGN           5         0
  02     FINAL-GROSS-SHIP       PACK   3       28                 VALUE SIGN           5         0
01       REPORT-RECORD          GRP    120     1                  VALUE
  02     REPORT-PLANT           A/N      3     6                  VALUE
  02     REPORT-ADDITIONS       NUM      5     14                 VALUE                5         0
  02     REPORT-DELETES         NUM      5     34                 VALUE                5         0
  02     REPORT-GROSS-SHIP      NUM      5     54                 VALUE                5         0
  02     REPORT-FINAL           A/N      5     64                 VALUE
```

LEVEL	is the level number or character that defines group items, hierarchy, and special entries in the Data Division.
NAME	is the section-name, data-name, and names of subordinate data items.
USAGE	is the form in which the data is stored.
BYTES	is the length.
POSN	is the position within the record. The first position is 1.
OCCUR	is the number of specified occurrences.
OCCLV	is the OCCURS nesting level.
REDEF	clause is indicated.
VALUE	clause is indicated.
SIGN	clause is indicated.
SYNC	clause is indicated.
DIGIT	the number of decimal digits represented by numeric items.
DECML	decimal places represented by numeric items.

4.3 Customization

The functions performed by SQA are provided as default values in the SQA macro set. Prior to executing SQA, you must determine which functions do not apply to your specific programming situations and/or local standards and then tailor SQA accordingly. There are two methods available for customizing SQA:

- The macro set was designed to anticipate all programming options that a programmer may or may not choose to implement. For this reason, there are cases where SQA will issue two or more contradictory NOTE diagnostics. The macro set can be modified by removing diagnostics that do not apply.

- SQA provides programming alternatives to override some of the macro set default values. These alternatives must be determined, and then implemented through the specification of control statements.

4.3.1 Macro Set Diagnostics

Contradictory diagnostics must be removed from the macro set. For example, in the case of the PERFORM statement, the following will be generated in the Input Listing for each applicable occurrence:

```
******* NOTE   N99   SQA44A 'PERFORM' WITHOUT THRU PHRASE
******* NOTE   N99   SQA45A 'PERFORM' WITH THRU PHRASE
```

All SQA diagnostics are identified with the diagnostic number in columns 73-77 of the macro set. Any diagnostic can be removed by locating the diagnostic number, and either deleting the lines of code or placing an asterisk in column 7 to comment them out; it is recommended that diagnostics be commented out. There will always be two or more lines of code, and in some cases they can be widely separated. Note that deleting the lines only removes the diagnostic: it does not delete the functional code.

In addition to SQA44A/SQA45A, other contradictory diagnostics are SQA14W/SQA15W and SQA48A/SQA49A. Locate the diagnostic numbers in columns 73-77 of the macro set, and select and remove the appropriate diagnostic.

4.3.2 Control Statements

Control statements override some macro set default values. They are coded as execution defaults in Area B (columns 12-72) following the macro set. The macro set then generates an Identification Division header as the last statement in the set, and a macro within the set removes all Identification Division headers except the first encountered (the one generated). This gives the control statements the appearance of being contained in the input source program.

They can also be embedded in the input source program; in this case, they only affect the portion of the program following the statement occurrence. Regardless of where they are placed, control statements do not take affect until translation begins.
Control statements are divided into two groups: those that specify coding standards and those that provide column formatting.

Coding Standards

Format:

```
$SQA-ABBR {Y}
          {N}
```

Specifies keyword abbreviations or full spellings:

Y

PIC, COMP, COMP-*n*, SYNC, and JUST.

N

PICTURE, COMPUTATIONAL, COMPUTATIONAL-*n*, SYNCHRONIZED, and JUSTIFIED.

Format:

```
             {YES}
$SQA-DLVERBS {Y  }
             {NO }
             {N  }
```

This option is only available under DSQA. It specifies how verbs are standardized:

Y or YES means that verbs are standardized as DL verbs.

N or NO means that verbs are standardized as COBOL verbs.

Format:

```
$SQA-EXTEN {Y}
           {N}
```

This command specifies whether IBM extension are NOTEd or ignored:

Y

IBM extensions retained for compatibility with the SYSTEM/360 compilers are NOTEd.

N

No diagnostics are produced.

Format:

```
                  {YES}
  $SQA-FROM/INTO  {Y  }
                  {NO }
                  {N  }
```

Specifies whether FROM/INTO statements are NOTEd or ignored:

Y

FROM/INTO statements are NOTEd. Y is an abbreviation for YES.

N

No diagnostics are produced. N is an abbreviation for NO.

Format:

```
      $SQA-INCR {n      }
                {2[ ODD]}
```

n

Can be specified as any number between 0 to 10 inclusive. If 0 is specified, level numbers are not changed.

2

Level numbers are incremented as 01, 02, 04, 06, 08, and so on.

2 ODD

Level numbers are incremented as 01, 03, 05, 07, 09, and so on.

Format:

```
               {data-name      [ [data-name      ]       ] }
      $SQA-MAP {section-name   [ [section-name   ] . . . ] }
               {procedure-name[ [procedure-name]        ] }
               {DATA           [ [               ]       ] }
```

This command defines the data areas to be mapped and listed in the Auxiliary Listing. A maximum of ten areas can be mapped in one translation (see Section 4.2.2). The Report Section and items defined within it cannot be mapped:

data-name

Data-names, all data items subordinate to them, and their associated attributes are listed.

section-name

Section-names, all data items subordinate to them, and their associated attributes are listed, where *section-name* represents FILE, WORKING-STORAGE, LINKAGE, or COMMUNICATION.

procedure-name

Procedure-names, all data items subordinate to them, and their associated attributes are listed, where *procedure-name* represents FILE, WORKING-STORAGE, LINKAGE, or COMMUNICATION.

DATA

A map of the entire Data Division, excluding the Report Section, is created.

Format:

```
$SQA-REDEF {Y}
           {N}
```

Specifies whether or not correction of the REDEFINES clause to ANSI specifications is to be made:

Y

The object of a REDEFINES clause is corrected.

N

No correction is made.

Format:

```
               {SHORT}
$SQA-RELATIONS {S    }
               {LONG }
               {L    }
```

This command enables you to set standards for relational operators in conditional statements:

S

and SHORT are synonyms. The long forms of relational operators are converted to short forms. For example, EQUAL TO becomes =, and GREATER THAN becomes >.

L

and LONG are synonyms. The short forms of relational operators are converted to long forms. For example, = becomes EQUAL TO, and > becomes GREATER THAN.

Format:

```
     $SQA-SEQN {nnnn}
               {10  }
```

Defines the increment for Procedure Division procedure-name numbering:

nn

Must be between 0 and 1000 inclusive. If 0 is specified, procedure-names are not changed.

Format:

```
   $SQA-SHORT {nn}
              {6 }
```

Defines the minimum length of an acceptable data-name or procedure-name:

nn

Must be between 1 and 30 inclusive.

Format:

```
   $SQA-SIGN {Y}
             {N}
```

This command specifies whether or not all COMPUTATIONAL, COMPUTATIONAL-3, and COMPUTATIONAL-4 items are to be signed:

Y

Signs are added where missing from PICTURE and VALUE clauses.

N

PICTURE and VALUE clauses are not changed.

Format:

```
$SQA-SIZE {Y}
          {N}
```

This command specifies numeric item size optimization:

Y

COMPUTATIONAL-3 data items are expanded to an odd number of digits except those with a length of 18, and COMPUTATIONAL and COMPUTATIONAL-4 data items are expanded to the number of digits necessary to fill the full amount of storage allocated to them: 4, 9, or 18.

N

Numeric item sizes are not changed.

Format:

```
$SQA-SYNC {Y}
          {N}
```

This command specifies the status of data item synchronization:

Y

A SYNCHRONIZED clause is appended to all elementary binary data item descriptions.

N

A warning is issued for each unsynchronized item if a dialect has been specified for any ANSI COBOL compiler earlier than the 1974 standard.

Format:

```
             {YES}
$SQA-VALUE {Y  }
             {NO }
             {N  }
```

Specifies whether values are assigned in the WORKING STORAGE Section:

Y

is an abbreviation for YES. Values are assigned in the WORKING STORAGE Section.

N

is an abbreviation for NO. Values are not assigned in the WORKING STORAGE Section.

Format:

```
$SQA-WORD {Y}
          {N}
```

Specifies whether or not optional keywords IS, USAGE, BY, ON, TIMES, KEY, CHARACTER, WHEN, and SIGN are to be included in the output text:

Y

Words are included.

N

Words are not included.

Format:

```
$SQA-77 {Y}
        {N}
```

Specifies whether or not level 77 data items are converted to level 01 items:

Y

All level 77 data items are converted.

N

No conversion is made.

Formatting Columns

SQA default values for formatting the columns in the output source program can be overridden with the following control statements. For each control statement, *integer* represents the column where the clause is to be aligned, and must be a number from 12 to 72 inclusive. Otherwise, the default is 0, indicating that no columnar alignment is to be executed.

Note: If the first word of the associated clause is placed so that it would extend beyond column 72, it will be right-aligned from column 72. By default, the clause is not aligned to a specific column.

Control Statement	**Assigns Columnar Alignment to**:
$SQA-BLANK-COLUMN *integer*	BLANK clause
$SQA-DEPENDING-COLUMN *integer*	DEPENDING clause
$SQA-INDEXED-COLUMN *integer*	INDEXED clause
$SQA-JUSTIFIED-COLUMN *integer*	JUSTIFIED clause
$SQA-KEY-COLUMN *integer*	ASCENDING/DESCENDING KEY clause
$SQA-OCCURS-COLUMN *integer*	OCCURS clause
$SQA-PICTURE-COLUMN *integer*	PICTURE clause
$SQA-REDEFINES-COLUMN *integer*	REDEFINES clause and RENAMES clause
$SQA-SIGN-COLUMN *integer*	SIGN clause
$SQA-SYNC-COLUMN *integer*	SYNCHRONIZED clause
$SQA-USAGE-COLUMN *integer*	USAGE clause
$SQA-VALUE-COLUMN *integer*	VALUE clause
$SQA-DEFAULT-COLUMN *integer*	All REDEFINES/RENAMES, OCCURS, DEPENDING, KEY, INDEXED, PICTURE, SIGN, BLANK, JUSTIFIED, USAGE, SYNCHRONIZED, and VALUE clauses that have not previously been assigned alignment by the specific control statements defined in this section.

5. VS COBOL II Quality Assurance (VQA) and DL VS COBOL II Quality Assurance (DVQA)

VS COBOL II Quality Assurance (VQA) provides functions for standardizing VS COBOL II source programs. DL VS COBOL II Quality Assurance (DVQA) provides functions for standardizing VS COBOL II source programs that contain DL verbs. Unless noted, the term VQA refers to both VS COBOL II Quality Assurance and DL VS COBOL II Quality Assurance.

The following sections contain descriptions of all functions performed by VQA and the methods available for customizing VQA to your needs.

5.1 Functions

Functions performed by VQA have either an *active* or *passive* effect on a VS COBOL II program. Active functions automatically change the source code; passive functions examine source code and provide you with a NOTE describing conditions that you may wish to change manually or with a macro.

NOTE diagnostics are listed in the Input or Output Listing in the form:

******* NOTE N99 VQA*nnz message*

VQA represents the VQA diagnostic abbreviation

nn represents the diagnostic number unique within VQA

z represents the diagnostic severity code

message represents a brief description of the diagnostic.

A complete explanation of NOTE diagnostics can be found in Appendix B.

The remainder of this section lists each VS COBOL II element examined and the action taken or condition NOTEd by VQA. The default action or issuance of a NOTE can be overridden via the control statement referenced or changed by removing the diagnostic within the macro set.

Note: All control statements begin with $VQA-. Control statements and diagnostic removal are described in Section 5.3.

5.1.1 Source Elements and Functions Performed

This section contains a list of each source element in each COBOL Division. Each source element is bolded, and the function it performs is listed directly below the element.

General

Data Definition Clause Order

Data definition clauses are always arranged in the sequence shown below. Note that RENAMES and REDEFINES are mutually exclusive and always placed first when coded; the other clauses do not have a pre-defined VS COBOL II precedence. ASCENDING/DESENDING KEY clauses are part of the OCCURS clause.

level number
data-name
RENAMES
REDEFINES
OCCURS
PICTURE
SIGN
BLANK WHEN ZERO
JUSTIFIED
USAGE
SYNCHRONIZED
VALUE

IBM Extensions

IBM extensions to the 1974 ANSI COBOL standard are NOTEd. See $VQA-EXTEN.

Environment Division

SAME AREA Clause

Two or more data sets allocated to the same I/O area via the SAME AREA clause are NOTEd; they impose restrictions on block size, record format, and program logic.

Data Division

General

Data Division file and record descriptions are *mapped* in the Auxiliary Listing. See $VQA-MAP in Section 5.3.2.

File Descriptions

BLOCK CONTAINS clauses specified for sequential files that do not contain a value of 0 and/or RECORD CONTAINS clauses containing a value of 0 are NOTEd.

Level Numbers

Data Definition level numbers are made consistent relative to the data hierarchy. See $VQA-INCR in Section 5.3.2.

Data-Names

Data-names shorter than the length specified by $VQA-SHORT are NOTEd. See $VQA-SHORT in Section 5.3.2.

Reserved Words

Abbreviations or full spellings are generated (e.g., PIC, COMP or PICTURE, COMPUTATIONAL). See $VQA-ABBR in Section 5.3.2.

Optional words IS, USAGE, BY, ON, TIMES, KEY, CHARACTER, WHEN, and SIGN are added to data attribute clauses (e.g., PICTURE IS, USAGE IS, etc.) or deleted. See $VQA-WORD in Section 5.3.2.

Group Items

Group item USAGE and SYNCHRONIZED clauses are moved to their subordinate elementary items.

Group FILLER Items

Group FILLER items referenced in the data hierarchy are NOTEd.

OCCURS...DEPENDING ON Clause

OCCURS...DEPENDING ON clauses within a record description can be confusing to the maintenance programmer. Nested OCCURS...DEPENDING ON clauses are NOTEd separately.

PICTURE and VALUE Clause

All numeric items that are COMPUTATIONAL, COMPUTATIONAL-3, or COMPUTATIONAL-4 are signed so the compiler does not generate additional instructions to remove and restore the signs. See $VQA-SIGN in Section 5.3.2.

All COMPUTATIONAL or COMPUTATIONAL-4 binary data items are changed to the number of digits necessary to fill the full amount of storage allocated to them, 4, 9, or 18. All COMPUTATIONAL-3 packed data items with an even number of digits in the picture, except those with a length of 18, are increased to an odd number of digits. See $VQA-SIZE in Section 5.3.2.

REDEFINES Clause

The REDEFINES object is made to conform to 1974 ANSI COBOL standards. If this action is overridden, non-conformance is NOTEd. See $VQA-REDEF in Section 5.3.2.

RENAMES Entry

RENAMES entries are NOTEd.

SYNCHRONIZED Clause

All binary, index, and floating point data items are SYNCHRONIZED to avoid the generation of alignment logic. If this action is overridden, non-synchronization is NOTEd. See $VQA-SYNC in Section 5.3.2.

VALUE Clause

VALUE IS SPACE or VALUE IS ZERO is added for Working-Storage items lacking a value other than those subordinate to a REDEFINES or OCCURS clause.

Procedure Division

Procedure-Names

Procedure-names shorter than the length specified by $VQA-SHORT are NOTEd as potentially non-meaningful. See $VQA-SHORT in Section 5.3.2.

Sequentially numbered prefixes are assigned to all procedure-name definitions and all references to them are altered. For example, the statement PERFORM MINOR-BREAK in the paragraph CHECK-FOR-BREAK becomes PERFORM 0400-MINOR-BREAK in the paragraph 0200-CHECK-FOR-BREAK. Instead of a cross-reference listing, the sequential prefix can then be used to locate the new logic path.

The following standards are used in numbering procedure-names:

- The resulting procedure-name will be a 4-digit number, a hyphen, and the first 25 characters of the original procedure-name. If the original name is actually longer than 25 characters and truncation causes the name to end with a hyphen, the character X is substituted for the hyphen.

- Procedure-names that are already numbered (1 to 4 digits followed by a hyphen) are re-sequenced by replacing the old number with a new number.

- Numeric procedure-names are suffixed with -PARA and then numbered; the original number is retained.

The increment for the paragraph sequence number is specified in $VQA-SEQN.

ACCEPT Verb

ACCEPT FROM CONSOLE is NOTEd as inefficient and prone to operator error.

ALTER Verb

A switch is added in Working-Storage for each ALTERed paragraph, and each ALTER statement is replaced with a MOVE statement that sets the switch. An IF statement testing the switch is then added prior to the GO TO statement in the ALTERed paragraph.

Arithmetic Statements

DISPLAY items and operands of inconsistent numeric usages and/or decimal alignments are NOTEd when encountered in arithmetic statements.

Conditional Statements

Group items used in an IF statement and data items of differing lengths that require padding before comparison are NOTEd.

CORRESPONDING Option

CORRESPONDING in an ADD, SUBTRACT, or MOVE statement is NOTEd as a potential maintenance problem.

Debugging Verbs

READY TRACE, RESET TRACE, EXHIBIT, and ON statements within production programs are NOTEd.

DISPLAY Verb

DISPLAY UPON CONSOLE is NOTEd as inefficient and prone to operator error.

GO TO Statements

A programmer-written or VQA-generated (in ALTER conversion) GO TO statement and its forward or backward direction are NOTEd.

INSPECT, EXAMINE, and TRANSFORM Verbs

Use of execution resources by these verbs is costly and therefore NOTEd.

I/O Statements

The use and/or failure to use FROM and INTO options is NOTEd.

ON SIZE ERROR Clause

NOTEd as inefficient and an indication of inadequate data editing.

PERFORM Verb and SORT Procedure Invocation

PERFORM verbs and SORT procedure invocations specified with or without the THRU option are NOTEd. The direction of the routines is NOTEd as either backward or forward in the program.

Program Segmentation

Overlay structures are NOTEd as having questionable value in a virtual system.

STOP Literal

NOTEd as prone to operator error, and invalidates the use of the program as a sub-program.

Subscripts/Indices

Subscripts used to accomplish sequential searching of a table are NOTEd; sequential searching of a table should be accomplished with a SEARCH statement. Subscripts defined as COMP-3 or DISPLAY data items are NOTEd as inefficient.

5.2 Reports

In addition to the Input and Output Listings generated by the Translator, VQA generates an Auxiliary Listing that contains a Summary Report and, if specified, a Map Report.

5.2.1 Summary Report

VQA automatically generates a Summary Report in the Auxiliary Listing. The VQA Summary Report provides a summary of all macro-generated diagnostics, as shown in the following example:

```
----------------------------------------------------------
|** COBOL QUALITY ASSURANCE DIAGNOSTIC SUMMARY           |
|**                                                     |
|** VQA39-'GO TO' VIOLATIONS                           10|
|** VQA40-FORWARD BRANCHES (GO TO)                      7|
|** VQA41-BACKWARD BRANCHES (GO TO)                     8|
|** VQA44-'PERFORM' OR 'SORT PROCEDURE' WITHOUT 'THRU'  6|
|** VQA45-'PERFORM' OR 'SORT PROCEDURE' WITH 'THRU'     7|
|** VQA46-FORWARD 'PERFORM' OR 'SORT PROCEDURE'         6|
|** VQA47-BACKWARD 'PERFORM' OR 'SORT PROCEDURE'        7|
|** VQA60A-SHORT DATA-NAMES OR PROCEDURE-NAMES          3|
|** VQA61-DATA-NAME TRUNCATIONS                         3|
|** VQA63-UNDEFINED PROCEDURE-NAMES                     1|
----------------------------------------------------------
```

Diagnostics can be modified or removed from the VQA macro set for specific programming needs. Refer to Section 5.3.1 for more information.

5.2.2 Map Report

Data areas to be mapped are specified by the $VQA-MAP control statement (refer to Section 5.3.2). Here is an example:

```
LEVEL  NAME            USAGE  BYTES  POSN  OCCUR  OCCLV REDEF  VALUE  SIGN  SYNC  DIGIT    DECML

       WORKING-STORAGE
01     WORK-AREA          GRP   80     1                         VALUE
  02   TRANSACTION-CODE   A/N   1      11                        VALUE
  02   PLANT              NUM   3      20                        VALUE              3         0
  02   PRODUCT-CODE       A/N   1      35                        VALUE
  02   MAKE-SHIP-CODE     A/N   1      56                        VALUE
  02   GROSS              NUM   5      66                        VALUE              5         0
01     COUNTERS           GRP   30     1                         VALUE
  02   PLANT-ADDITIONS    PACK  3      1                         VALUE
  02   PLANT-CHANGES      PACK  3      4                         VALUE  SIGN        5         0
  02   PLANT-DELETES      PACK  3      7                         VALUE  SIGN        5         0
  02   PLANT-GROSS-MADE   PACK  3      10                        VALUE  SIGN        5         0
  02   PLANT-GROSS-SHIP   PACK  3      13                        VALUE  SIGN        5         0
  02   FINAL-ADDITIONS    PACK  3      16                        VALUE  SIGN        5         0
  02   FINAL-CHANGES      PACK  3      19                        VALUE  SIGN        5         0
  02   FINAL-DELETES      PACK  3      22                        VALUE  SIGN        5         0
  02   FINAL-GROSS-MADE   PACK  3      25                        VALUE  SIGN        5         0
  02   FINAL-GROSS-SHIP   PACK  3      28                        VALUE  SIGN        5         0
01     REPORT-RECORD      GRP   120    1                         VALUE
  02   REPORT-PLANT       A/N     3    6                         VALUE
  02   REPORT-ADDITIONS   NUM     5    14                        VALUE              5         0
  02   REPORT-DELETES     NUM     5    34                        VALUE              5         0
  02   REPORT-GROSS-SHIP  NUM     5    54                        VALUE              5         0
  02   REPORT-FINAL       A/N     5    64                        VALUE
```

LEVEL is the level number or character that defines group items, hierarchy, and special entries in the Data Division.

NAME is the section-name, data-name, and names of subordinate data items.

USAGE is the form in which the data is stored.

BYTES is the length.

POSN is the position within the record. The first position is 1.

OCCUR is the number of specified occurrences.

OCCLV is the OCCURS nesting level.

REDEF clause is indicated.

VALUE clause is indicated.

SIGN clause is indicated.

SYNC clause is indicated.

DIGIT the number of decimal digits represented by numeric items.

DECML decimal places represented by numeric items.

5.3 Customization

The functions performed by VQA are provided as default values in the VQA macro set. Prior to executing VQA, you must determine which functions do not apply to your specific programming situations and/or local standards and then tailor VQA accordingly. There are two methods available for customizing VQA:

- The macro set was designed to anticipate all programming options that a programmer may or may not choose to implement. For this reason, there are cases where VQA will issue two or more contradictory NOTE diagnostics. The macro set can be modified by removing diagnostics that do not apply.
- VQA provides programming alternatives to override some of the macro set default values. These alternatives must be determined, and then implemented through the specification of control statements.

5.3.1 Macro Set Diagnostics

Contradictory diagnostics must be removed from the macro set. For example, in the case of the PERFORM statement, the following will be generated in the Input Listing for each applicable occurrence:

```
******* NOTE   N99   VQA44A 'PERFORM' WITHOUT THRU PHRASE
******* NOTE   N99   VQA45A 'PERFORM' WITH THRU PHRASE
```

All VQA diagnostics are identified with the diagnostic number in columns 73-77 of the macro set. Any diagnostic can be removed by locating the diagnostic number, and either deleting the lines of code or placing an asterisk in column 7 to comment them out; it is recommended that diagnostics be commented out. There will always be two or more lines of code, and in some cases they can be widely separated. Note that deleting the lines only removes the diagnostic: it does not delete the functional code.

In addition to VQA44A/VQA45A, other contradictory diagnostics are VQA14W/VQA15W and VQA48A/VQA49A. Locate the diagnostic numbers in columns 73-77 of the macro set, and select and remove the appropriate diagnostic.

5.3.2 Control Statements

Control statements override some macro set default values. They are coded as execution defaults in Area B (columns 12-72) following the macro set. The macro set then generates an Identification Division header as the last statement in the set, and a macro within the set removes all Identification Division headers except the first encountered (the one generated). This gives the control statements the appearance of being contained in the input source program.

They can also be embedded in the input source program; in this case, they only affect the portion of the program following the statement occurrence. Regardless of where they are placed, control statements do not take affect until translation begins.

Control statements are divided into two groups: those that specify coding standards and those that provide column formatting.

Coding Standards

VQA functions that affect VS COBOL II coding standards ensure that the code is reasonably efficient and that it does not contain obsolete language elements. Where indicated, macro set default values for these types of functions are shown as underlined parameters of the following control statements and can be overridden by specifying the alternative.

Format:

```
$VQA-ABBR {Y}
          {N}
```

This command specifies keyword abbreviations or full spellings:

Y

PIC, COMP, COMP-*n*, SYNC, and JUST.

N

PICTURE, COMPUTATIONAL, COMPUTATIONAL-*n*, SYNCHRONIZED, and JUSTIFIED.

Format:

```
              {YES}
$VQA-DLVERBS {Y  }
              {NO }
              {N  }
```

This option is only available under DVQA. It specifies how verbs are standardized:

Y

or YES means that verbs are standardized as DL verbs.

N

or NO means that verbs are standardized as VS COBOL II verbs.

Format:

```
$VQA-EXTEN {Y}
           {N}
```

This command specifies whether IBM extension are NOTEd or ignored:

Y

IBM extensions retained for compatibility with the SYSTEM/360 compilers are NOTEd.

N

No diagnostics are produced.

Format:

```
               {YES}
$VQA-FROM/INTO {Y  }
               {NO }
               {N  }
```

This command specifies whether FROM/INTO statements are NOTEd or ignored:

Y

FROM/INTO statements are NOTEd. Y is an abbreviation for YES.

N

No diagnostics are produced. N is an abbreviation for NO.

Format:

```
$VQA-INCR {n      }
          {2[ ODD]}
```

This command specifies the level numbering increment to be used in the Data Division.

n

Can be specified as any number from 0 to 10 inclusive. If 0 is specified, level numbers are not changed.

2

Level numbers are incremented as 01, 02, 04, 06, 08, and so on.

2 ODD

Level numbers are incremented as 01, 03, 05, 07, 09, and so on.

Format:

```
         {data-name    [ [data-name    ]       ] }
$VQA-MAP {section-name[ [section-name] . . . ] }
         {DATA         [ [             ]       ] }
```

This command defines the data areas to be mapped and listed in the Auxiliary Listing. A maximum of ten areas can be mapped in one translation (see Section 5.2.2). The Report Section and items defined within it cannot be mapped.

data-name
Data-names, all data items subordinate to them, and their associated attributes are listed.

section-name
Section-names, all data items subordinate to them, and their associated attributes are listed, where *section-name* represents FILE, WORKING-STORAGE, LINKAGE, or COMMUNICATION.

DATA
A map of the entire Data Division, excluding the Report Section, is created.

Format:

```
$VQA-REDEF {Y}
           {N}
```

Specifies whether or not correction of the REDEFINES clause to ANSI specifications is to be made:

Y
The object of a REDEFINES clause is corrected.

N
No correction is made.

Format:

```
               {SHORT}
$VQA-RELATIONS {S    }
               {LONG }
               {L    }
```

This command enables you to set standards for relational operators in conditional statements:

S

and SHORT are synonyms. The long forms of relational operators are converted to short forms. For example, EQUAL TO becomes =, and GREATER THAN becomes >.

L

and LONG are synonyms. The short forms of relational operators are converted to long forms. For example, = becomes EQUAL TO, and > becomes GREATER THAN.

Format:

```
$VQA-SEQN {nnnn}
          {10  }
```

This command defines the increment for Procedure Division procedure-name numbering:

nn

Must be from 0 to 1000 inclusive. If 0 is specified, procedure-names are not changed.

Format:

```
$VQA-SHORT {nn}
           {6 }
```

This command defines the minimum length of an acceptable data-name or procedure-name:

nn

Must be between 1 and 30 inclusive.

Format:

```
$VQA-SIGN {Y}
          {N}
```

This command specifies whether or not all COMPUTATIONAL, COMPUTATIONAL-3, and COMPUTATIONAL-4 items are to be signed:

Y

Signs are added where missing from PICTURE and VALUE clauses.

N

PICTURE and VALUE clauses are not changed.

Format:

```
$VQA-SIZE {Y}
          {N}
```

This command specifies numeric item size optimization:

Y

COMPUTATIONAL-3 data items are expanded to an odd number of digits except those with a length of 18, and COMPUTATIONAL and COMPUTATIONAL-4 data items are expanded to the number of digits necessary to fill the full amount of storage allocated to them: 4, 9, or 18.

N

Numeric item sizes are not changed.

Format:

```
$VQA-SYNC {Y}
          {N}
```

This command specifies the status of data item synchronization:

Y

A SYNCHRONIZED clause is appended to all elementary binary data item descriptions.

N

A warning is issued for each unsynchronized item if a dialect has been specified for any ANSI COBOL compiler earlier than the 1974 standard.

Format:

```
             {YES}
$VQA-THRU    {Y   }
             {NO  }
             {N   }
```

Specifies whether THRU statements are NOTEd or ignored:

Y

THRU statements are NOTEd. Y is an abbreviation for YES.

N

No diagnostics are produced. N is an abbreviation for NO.

Format:

```
              {YES}
$VQA-VALUE    {Y   }
              {NO  }
              { N  }
```

Specifies whether values are assigned in the WORKING STORAGE Section:

Y is an abbreviation for YES. Values are assigned in the WORKING STORAGE Section.

N is an abbreviation for NO. Values are not assigned in the WORKING STORAGE Section.

Format:

```
$VQA-WORD {Y}
          {N}
```

This command specifies whether or not optional keywords IS, USAGE, BY, ON, TIMES, KEY, CHARACTER, WHEN, and SIGN are to be included in the output text:

Y

Words are included.

N

Words are not included.

Format:

```
$VQA-77 {Y}
        {N}
```

This command specifies whether or not level 77 data items are converted to level 01 items:

Y

All level 77 data items are converted.

N

No conversion is made.

Formatting Columns

VQA default values for formatting the columns in the output source program can be overridden with the following control statements. For each control statement, *integer* represents the column where the clause is to be aligned, and must be a number from 12 to 72 inclusive. Otherwise, the default is 0, indicating that no columnar alignment is to be executed.

Note: If the first word of the associated clause is placed so that it would extend beyond column 72, it will be right-aligned from column 72. By default, the clause is not aligned to a specific column.

Control Statement	**Assigns Columnar Alignment to:**
$VQA-BLANK-COLUMN *integer*	BLANK clause
$VQA-DEPENDING-COLUMN *integer*	DEPENDING clause
$VQA-INDEXED-COLUMN *integer*	INDEXED clause
$VQA-JUSTIFIED-COLUMN *integer*	JUSTIFIED clause
$VQA-KEY-COLUMN *integer*	ASCENDING/DESCENDING KEY clause
$VQA-OCCURS-COLUMN *integer*	OCCURS clause
$VQA-PICTURE-COLUMN *integer*	PICTURE clause
$VQA-REDEFINES-COLUMN *integer*	REDEFINES clause and RENAMES clause
$VQA-SIGN-COLUMN *integer*	SIGN clause
$VQA-SYNC-COLUMN *integer*	SYNCHRONIZED clause
$VQA-USAGE-COLUMN *integer*	USAGE clause
$VQA-VALUE-COLUMN *integer*	VALUE clause
$VQA-DEFAULT-COLUMN *integer*	All REDEFINES/RENAMES, OCCURS, DEPENDING, KEY, INDEXED, PICTURE, SIGN, BLANK, JUSTIFIED, USAGE, SYNCHRONIZED, and VALUE clauses that have not previously been assigned alignment by the specific control statements defined in this section.

Appendix A. Quality Assurance (QA) Facility Example

This appendix provides an example of the Input, Output, and Auxiliary Listings produced as a result of CA-MetaCOBOL+ Translator translation under control of the QA Facility CQA macro set. Each page of the Output Listing is shown opposite its corresponding Input Listing page.

Input Listing

```
CA-METACOBOL+ INPUT      V1.1   01/04/92 13:04  PAGE 1

                   CA-METACOBOL+ IS LEASED FROM
             COMPUTER ASSOCIATES INTERNATIONAL, INC.
             COPYRIGHT (C) 1989.  USE OF THIS PROGRAM
              BY UNAUTHORIZED PERSONS IS PROHIBITED.

OPTION LISTIN,LISTOUT,RESEQ=NUM,ID=*MC
OPTION DIALECT=YO

**INSTALLATION SPECIFICATIONS:
ACCT=ENQ   CRSTACK=0  DATE=MDY   DDID=000   DEPTH=66
DIALECT=WO ID=*MC     LIBCORE=0  LIBE=YES   LIBPASS=NO
MBLK=14000 OBLK=14000 PRESERV=(SQL)         PSTAT=
PVER=000   RESEQ=NUM  SAMPG=YES  SOURCE=A   STACK=3200
TERM=YES
FORMAT=OPEN
OPTION=(LISTOUT,SEPPAR,DECK,LISTIN,APOST,NOISEQ,COMMENT,NOINVDEC)

**OPTIONS IN EFFECT:
CONVERT=YES,,TARGET=A,,,,,,,
TERM=YES,,,FREE=,,,,PSTAT=
,PVER=000,FORMAT=(EP,EV,DFC40,DTT40,DI3X6,
PC,PP,PI),,NOSYNTAX

**OPTIONS IN EFFECT:
APOST         NOCLOCK       COMMENT       COPY=PASSIVE
CRSTACK=0     DDID=000      DECK          DEPTH=66
DIALECT=WO    ENABLE=0      ID=*MC        NOIGNORE
NO-INC        NOINVDEC      NOISEQ        IXIT=
LCORE=0       NOLISTALL     LISTIN        LISTOUT
NOLSEQ        NOTE=I        PRESERVE=SQL  PSTAT=
PVER=000      REPLACE=ACTIVE              RESEQ=NUM   SEPPAR
SOURCE=A      NOSUPPRESS    NOTERM        NOTRACE
UPSI1=        UPS12=        UPS13=        UPS14=
UPSI5=        UPS16=        UPS17=        UPS18=
CASE=UPPERCASE              KEYWORDS=UPPERCASE
              NAMES=UPPERCASE
VAR=
FORMAT=(IC,EC,EV,ESC31,E12,DFC31,DT31,DI2X5,PC,PP,PI2X4,PVBCDEH)

00001           W    X1 : INDEX-1
00002           W    X2 : INDEX-2
00003           W    X3 : INDEX-3
00004           W    PREV-PROGRAM-ID : PREVIOUS-PROGRAM-ID
00005           W    PREV-CATEGORY : PREVIOUS-CATEGORY
00006           W    PREV-AUTHOR : PREVIOUS-AUTHOR
00007           W    A-YEAR : ACCEPT-YEAR
00008           W    A-MONTH : ACCEPT-MONTH
00009           W    A-DAY : ACCEPT-DAY
00010           W    D-MONTH : DECODED-MONTH
00011           W    D-DAY : DECODED-DAY
00012           W    D-YEAR : DECODED-YEAR
00013           *$LIBED CQA
00014 L   000010*$HDR CQA     COBOL QUALITY ASSURANCE-V1.0
00015 L   000020*$NOLIST
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 2

04848 L   %      IDENTIFICATION DIVISION.
04849                $CQA-INCR  2 ODD
04850                $CQA-77    'Y'
04851                $CQA-ABBR  'N'
04852                $CQA-WORD  'N'
04853                $CQA-SYNC  'Y'
04854                $CQA-SIZE  'Y'
04855                $CQA-SIGN  'Y'
04856                $CQA-SEQN  10
04857                $CQA-SHORT 2
04858                $CQA-REDEFINES-COLUMN 31
04859                $CQA-DEPENDING-COLUMN 31
04860                $CQA-KEY-COLUMN       31
04861                $CQA-PICTURE-COLUMN   40
04862                $CQA-INDEXED-COLUMN   31
04863                $CQA-SIGN-COLUMN      51
04864                $CQA-BLANK-COLUMN     51
04865                $CQA-JUSTIFIED-COLUMN 51
04866                $CQA-USAGE-COLUMN     51
04867                $CQA-SYNC-COLUMN      59
04868                $CQA-VALUE-COLUMN     59
04869     000700 IDENTIFICATION DIVISION.
04870     000710 PROGRAM-ID. AMR.
04871     000720 AUTHOR. COMPUTER ASSOCIATES, INC.
04872     000730 ENVIRONMENT DIVISION.
04873     000740 CONFIGURATION SECTION.
04874     000750 SOURCE-COMPUTER. IBM-370.
04875     000770 OBJECT-COMPUTER. IBM-370.
04876     000900 SPECIAL-NAMES.
04877     000910     C01 IS TOP-OF-PAGE.
04878     000920 INPUT-OUTPUT SECTION.
04879     000930 FILE-CONTROL.
04880     000940     SELECT ACCT-FILE ASSIGN UT-S-SYSACCT.
04881     000960     SELECT TRANS-FILE    ASSIGN TO UT-S-SYSORT.
04882     000980     SELECT OLD-FILE
04883     000990     ASSIGN TO UT-S-SYSOLD.
04884     001000     SELECT NEW-FILE
04885     001010              ASSIGN UT-S-SYSNEW.
04886     001020     SELECT REPORT-FILE
04887     001030           ASSIGN TO UT-S-SYSPRINT.
04888     001200 DATA DIVISION.
04889     001210 FILE SECTION.
04890     001220 FD  ACCT-FILE RECORDING MODE F LABEL RECORDS STANDARD
04891     001241                       BLOCK CONTAINS 0 RECORDS
04892     001250          DATA RECORD ACCT-RECORD.
04893     001260 01  ACCT-RECORD.
04894     001270     02 FILLER   PIC X.
04895     001280     02 ACCT-KEY   PIC X(4).
04896     001290        02 FILLER          PIC X.
04897     001300     02 ACCT-CATEGORY OCCURS 38 TIMES INDEXED BY X1 PIC X.
04898     001310     02 FILLER PIC X.
04899     001320     02 ACCT-DATE.
04900     001330       03 ACCT-MO  PIC XX.
04901     001340         03 FILLER    PIC X.
04902     001350       03 ACCT-DY       PIC XX.
04903     001360       03 FILLER        PICTURE X.
04904     001370       03 ACCT-YR       PIC XX.
04905     001380     02 FILLER          PIC X.
04906     001390     02 ACCT-PROGRAM-ID PICTURE X(10).
04907     001400       02 FILLER          PIC X.
04908     001410     02 ACCT-AUTHOR     PIC X(15).
04909     001470 SD  TRANS-FILE
04910     001490                                  DATA RECORD TRANS-RECORD.
04911     001500 01  TRANS-RECORD.
04912     001510     02 TRANS-KEY PIC X(77).
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 1    AMR

00001     000010 IDENTIFICATION DIVISION.
00002     000020 PROGRAM-ID. AMR.
00003     000030 AUTHOR. COMPUTER ASSOCIATES, INC.
00004     000040 ENVIRONMENT DIVISION.
00005     000050 CONFIGURATION SECTION.
00006     000060 SOURCE-COMPUTER. IBM-370.
00007     000070 OBJECT-COMPUTER. IBM-370.
00008     000080 SPECIAL-NAMES. C01 IS TOP-OF-PAGE.
00009     000090 INPUT-OUTPUT SECTION.
00010     000100 FILE-CONTROL.
00011     000110     SELECT ACCT-FILE ASSIGN UT-S-SYSACCT.
00012     000120     SELECT TRANS-FILE ASSIGN TO UT-S-SYSORT.
00013     000130     SELECT OLD-FILE ASSIGN TO UT-S-SYSOLD.
00014     000140     SELECT NEW-FILE ASSIGN UT-S-SYSNEW.
00015     000150     SELECT REPORT-FILE ASSIGN TO UT-S-SYSPRINT.
00016     000160 DATA DIVISION.
00017     000170 FILE SECTION.
00018     000180 FD  ACCT-FILE
00019     000190                                 RECORDING MODE F
00020     000200                                 LABEL RECORDS STANDARD
00021     000210                                 BLOCK CONTAINS 0 RECORDS
00022     000220                                 DATA RECORD ACCT-RECORD.
00023     000230 01  ACCT-RECORD.
00024     000240     03 FILLER                   PICTURE X.
00025     000250     03 ACCT-KEY                 PICTURE X(4).
00026     000260     03 FILLER                   PICTURE X.
00027     000270     03 ACCT-CATEGORY            OCCURS 38
00028     000280                        INDEXED INDEX-1
00029     000290                                 PICTURE X.
00030     000300     03 FILLER                   PICTURE X.
00031     000310     03 ACCT-DATE.
00032     000320        05 ACCT-MO               PICTURE XX.
00033     000330        05 FILLER                PICTURE X.
00034     000340        05 ACCT-DY               PICTURE XX.
00035     000350        05 FILLER                PICTURE X.
00036     000360        05 ACCT-YR               PICTURE XX.
00037     000370     03 FILLER                   PICTURE X.
00038     000380     03 ACCT-PROGRAM-ID          PICTURE X(10).
00039     000390     03 FILLER                   PICTURE X.
00040     000400     03 ACCT-AUTHOR              PICTURE X(15).
00041     000410 SD  TRANS-FILE
00042     000420                                 DATA RECORD TRANS-RECORD.
00043     000430 01  TRANS-RECORD.
00044     000440     03 TRANS-KEY                PICTURE X(77).
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 3

04913     001520     02 FILLER       PIC X(7).
04914     001590 FD  OLD-FILE
04915     001600               RECORDING MODE F
04916     001610           LABEL RECORDS STANDARD
04917     001611                  BLOCK CONTAINS 0 RECORDS
04918     001620       DATA RECORD OLD-RECORD.
04919     001630 01  OLD-RECORD     PICTURE X(80).
04920     001690 FD  NEW-FILE
04921     001700                        RECORDING MODE F
04922     001710                        LABEL RECORDS STANDARD
04923     001711                        BLOCK CONTAINS 0 RECORDS
04924     001720                        DATA RECORD NEW-RECORD.
04925     001730 01  NEW-RECORD.
04926     001740     02 NEW-KEY.
04927     001750     03 NEW-SUB-KEY.
04928     001760     04 NEW-CODE    PIC X.
04929     001770       04 NEW-CATEGORY-1 PICTURE X(38).
04930     001780      04 NEW-AUTHOR-1 PIC X(15).
04931     001790      04 NEW-PROGRAM-ID-1 PIC X(10).
04932     001800       03 NEW-R-KEY  REDEFINES NEW-SUB-KEY.
04933     001810       04 FILLER         PIC X.
04934     001820       04 NEW-AUTHOR-2 PIC X(15).
04935     001830         04 NEW-PROGRAM-ID-2 PIC X(10).
04936     001840         04 NEW-CATEGORY-2 PIC X(38).
04937     001850      03 NEW-DATE.
04938     001860       04 NEW-YR      PIC XX.
04939     001870         04 NEW-MO      PIC XX.
04940     001880         04 NEW-DY      PIC XX.
04941     001890     02 NEW-RUNS   COMP-3 PIC S9(5).
04942     001900       02 NEW-EXCEPTIONS  PIC S9(7) COMP-3.
04943     001910     02 FILLER          PICTURE X(3).
04944     001970 FD  REPORT-FILE RECORDING MODE F
04945     001990                 LABEL RECORDS OMITTED
04946     001991                        BLOCK CONTAINS 0 RECORDS
04947     002000                    DATA RECORD REPORT-RECORD.
04948     002010 01  REPORT-RECORD. 02 FILLER PIC X(121).
04949     002110 WORKING-STORAGE SECTION.
04950     002120 77 ACCT-REC-COUNT PIC S9(7) VALUE +0 COMP-3.
04951     002130 77  REPORT-CODE     PICTURE X VALUE LOW-VALUE.
04952     002140 77 DEPTH-COUNTER  PIC S9(3) COMP-3 VALUE +57.
04953     002150 77  REPORT-PAGE PIC S9(3) COMP-3 VALUE +0.
04954     002160 77  PREV-PROGRAM-ID  PIC X(10).
04955     002170 77  PREV-CATEGORY     PIC X(38).
04956     002180 77  PREV-AUTHOR PIC X(15).
04957     002190 77  FORM-CONTROL PIC X.
04958     002200 01  ACCEPT-DATE.
04959     002210     02 A-YEAR  PIC 99.
04960     002220     02 A-MONTH    PIC 99.
04961     002230     02 A-DAY      PIC 99.
04962     002240 01  DECODED-DATE.
04963     002250     02 D-MONTH     PIC 99.
04964     002260     02 D-DAY           PIC 99.
04965     002270     02 D-YEAR          PIC 99.
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 2    AMR

00045     000450     03 FILLER                   PICTURE X(7).
00046     000460 FD  OLD-FILE
00047     000470                                 RECORDING MODE F
00048     000480                                 LABEL RECORDS STANDARD
00049     000490                                 BLOCK CONTAINS 0 RECORDS
00050     000500                                 DATA RECORD OLD-RECORD.
00051     000510 01  OLD-RECORD                  PICTURE X(80).
00052     000520 FD  NEW-FILE
00053     000530                                 RECORDING MODE F
00054     000540                                 LABEL RECORDS STANDARD
00055     000550                                 BLOCK CONTAINS 0 RECORDS
00056     000560                                 DATA RECORD NEW-RECORD.
00057     000570 01  NEW-RECORD.
00058     000580     03 NEW-KEY.
00059     000590        05 NEW-SUB-KEY.
00060     000600           07 NEW-CODE           PICTURE X.
00061     000610           07 NEW-CATEGORY-1     PICTURE X(38).
00062     000620           07 NEW-AUTHOR-1       PICTURE X(15).
00063     000630           07 NEW-PROGRAM-ID-1   PICTURE X(10).
00064     000640        05 NEW-R-KEY    REDEFINES NEW-SUB-KEY.
00065     000650           07 FILLER             PICTURE X.
00066     000660           07 NEW-AUTHOR-2       PICTURE X(15).
00067     000670           07 NEW-PROGRAM-ID-2   PICTURE X(10).
00068     000680           07 NEW-CATEGORY-2     PICTURE X(38).
00069     000690        05 NEW-DATE.
00070     000700           07 NEW-YR             PICTURE XX.
00071     000710           07 NEW-MO             PICTURE XX.
00072     000720           07 NEW-DY             PICTURE XX.
00073     000730     03 NEW-RUNS                 PICTURE S9(5)
00074     000740                                          COMPUTATIONAL-3.
00075     000750     03 NEW-EXCEPTIONS           PICTURE S9(7)
00076     000760                                          COMPUTATIONAL-3.
00077     000770     03 FILLER                   PICTURE X(3).
00078     000780 FD  REPORT-FILE
00079     000790                                 RECORDING MODE F
00080     000800                                 LABEL RECORDS OMITTED
00081     000810                                 BLOCK CONTAINS 0 RECORDS
00082     000820                                 DATA RECORD REPORT-RECORD.
00083     000830 01  REPORT-RECORD.
00084     000840     03 FILLER                   PICTURE X(121).
00085     000850 WORKING-STORAGE SECTION.
00086     000860 01  ACCT-REC-COUNT              PICTURE S9(7)
00087     000870                                            COMPUTATIONAL-3
00088     000880                                                  VALUE +0.
00089     000890 01  REPORT-CODE             PICTURE X          VALUE
00090     000900                                 LOW-VALUE.
00091     000910 01  DEPTH-COUNTER           PICTURE S999
00092     000920                                            COMPUTATIONAL-3
00093     000930                                                VALUE +57.
00094     000940 01  REPORT-PAGE                 PICTURE S999
00095     000950                                            COMPUTATIONAL-3
00096     000960                                                 VALUE +0.
00097     000970 01  PREVIOUS-PROGRAM-ID     PICTURE X(10)    VALUE SPACES.
00098     000980 01  PREVIOUS-CATEGORY       PICTURE X(38)    VALUE SPACES.
00099     000990 01  PREVIOUS-AUTHOR         PICTURE X(15)    VALUE SPACES.
00100     001000 01  FORM-CONTROL            PICTURE X        VALUE SPACES.
00101     001010 01  ACCEPT-DATE.
00102     001020     03 ACCEPT-YEAR          PICTURE 99         VALUE ZERO.
00103     001030     03 ACCEPT-MONTH         PICTURE 99         VALUE ZERO.
00104     001040     03 ACCEPT-DAY           PICTURE 99         VALUE ZERO.
00105     001050 01  DECODED-DATE.
00106     001060     03 DECODED-MONTH        PICTURE 99         VALUE ZERO.
00107     001070     03 DECODED-DAY          PICTURE 99         VALUE ZERO.
00108     001080     03 DECODED-YEAR         PICTURE 99         VALUE ZERO.
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 4

04966     002280 01  AMERICAN-DATE-FORMAT REDEFINES DECODED-DATE PIC 9(6).
04967     002350 01  SORT-RECORD.
04968     002360     02 SORT-KEY.
04969     002370     03 SORT-SUB-KEY.
04970     002380       04 SORT-CODE   PIC X.
04971     002390         04 SORT-CATEGORY-1 PIC X(38).
04972     002400       04 SORT-AUTHOR-1 PIC X(15).
04973     002410        04 SORT-PROGRAM-ID-1 PIC X(10).
04974     002420       03 SORT-R-KEY  REDEFINES SORT-SUB-KEY.
04975     002430       04 FILLER      PIC X.
04976     002440       04 SORT-AUTHOR-2 PIC X(15).
04977     002450       04 SORT-PROGRAM-ID-2 PICTURE X(10).
04978     002460       04 SORT-CATEGORY-2 PIC X(38).
04979     002470         03 SORT-DATE     PIC X(6).
04980     002480       03 SORT-SEQ      PICTURE 9(7).
04981     002490     02 SORT-RUNS    PIC S9(5) COMP-3 VALUE +0.
04982     002500     02 SORT-EXCEPTIONS PIC S9(7) COMP-3 VALUE +0.
04983     002570 01  WORK-CATEGORY.
04984     002580    02 WORK-CHARACTER  OCCURS 38 TIMES INDEXED BY X3 PIC X.
04985     002590     02 WORK-DATE.
04986     002600       03 WORK-YR       PIC XX.
04987     002610       03 WORK-MO       PIC XX.
04988     002620       03 WORK-DY       PIC XX.
04989     002630     02 WORK-NUMERIC    PIC 9(7).
04990     002640     02 WORK-EXCEPTIONS REDEFINES WORK-NUMERIC.
04991     002650       03 WORK-BYTE     OCCURS 7 TIMES INDEXED BY X2 PIC X.
04992     002720 01  OLD-WORK.
04993     002730     02 OLD-KEY.
04994     002740     03 OLD-SUB-KEY   PIC X(64).
04995     002750     03 FILLER  PIC X(6).
04996     002760     02 FILLER          PIC X(10).
04997     002830 01  STACK-AREA.
04998     002840     02 STACK-KEY.
04999     002850     03 STACK-SUB-KEY.
05000     002860       04 STACK-CODE  PIC X.
05001     002870         04 STACK-CATEGORY-1 PIC X(38).
05002     002880       04 STACK-AUTHOR-1 PIC X(15).
05003     002890         04 STACK-PROGRAM-ID-1 PIC X(10).
05004     002900     03  STACK-R-KEY      REDEFINES STACK-SUB-KEY.
05005     002910         04 FILLER PIC X.
05006     002920      04 STACK-AUTHOR-2 PIC X(15).
05007     002930        04 STACK-PROGRAM-ID-2 PIC X(10).
05008     002940         04 STACK-CATEGORY-2 PIC X(38).
05009     002950       03 STACK-DATE    PIC X(6).
05010     002960     02 STACK-RUNS      PIC S9(5) COMP-3 VALUE +0.
05011     002970     02 STACK-EXCEPTIONS PIC S9(7) COMP-3 VALUE +0.
05012     002980     02 FILLER          PIC X(3) VALUE SPACE.
05013     003050 01 REPORT-HEAD-1.
05014     003060     02 FILLER  PIC X(10) VALUE SPACES.
05015     003070     02 REPORT-HEAD-DATE PIC XX/XX/XX.
05016     003080     02 FILLER      PIC X(10) VALUE SPACE.
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 3    AMR

00109     001090 01  AMERICAN-DATE-FORMAT
00110     001100                        REDEFINES DECODED-DATE
00111     001110                                 PICTURE 9(6).
00112     001120 01  SORT-RECORD.
00113     001130     03 SORT-KEY.
00114     001140        05 SORT-SUB-KEY.
00115     001150           07 SORT-CODE      PICTURE X        VALUE SPACES.
00116     001160           07 SORT-CATEGORY-1  PICTURE X(38)  VALUE SPACES.
00117     001170           07 SORT-AUTHOR-1   PICTURE X(15)   VALUE SPACES.
00118     001180          07 SORT-PROGRAM-ID-1  PICTURE X(10) VALUE SPACES.
00119     001190        05 SORT-R-KEY   REDEFINES SORT-SUB-KEY.
00120     001200           07 FILLER             PICTURE X.
00121     001210           07 SORT-AUTHOR-2      PICTURE X(15).
00122     001220           07 SORT-PROGRAM-ID-2  PICTURE X(10).
00123     001230           07 SORT-CATEGORY-2    PICTURE X(38).
00124     001240        05 SORT-DATE             PICTURE X(6) VALUE SPACES.
00125     001250        05 SORT-SEQ              PICTURE 9(7)   VALUE ZERO.
00126     001260     03 SORT-RUNS                PICTURE S9(5)
00127     001270                                            COMPUTATIONAL-3
00128     001280                                                VALUE +0.
00129     001290     03 SORT-EXCEPTIONS          PICTURE S9(7)
00130     001300                                            COMPUTATIONAL-3
00131     001310                                                VALUE +0.
00132     001320 01  WORK-CATEGORY.
00133     001330     03 WORK-CHARACTER           OCCURS 38
00134     001340                        INDEXED INDEX-3
00135     001350                                 PICTURE X.
00136     001360     03 WORK-DATE.
00137     001370        05 WORK-YR               PICTURE XX   VALUE SPACES.
00138     001380        05 WORK-MO               PICTURE XX   VALUE SPACES.
00139     001390        05 WORK-DY               PICTURE XX   VALUE SPACES.
00140     001400     03 WORK-NUMERIC             PICTURE 9(7) VALUE ZERO.
00141     001410     03 WORK-EXCEPTIONS REDEFINES WORK-NUMERIC.
00142     001420        05 WORK-BYTE             OCCURS 7
00143     001430                        INDEXED INDEX-2
00144     001440                                 PICTURE X.
00145     001450 01  OLD-WORK.
00146     001460     03 OLD-KEY.
00147     001470        05 OLD-SUB-KEY       PICTURE X(64)    VALUE SPACES.
00148     001480        05 FILLER             PICTURE X(6)    VALUE SPACES.
00149     001490     03 FILLER                PICTURE X(10)   VALUE SPACES.
00150     001500 01  STACK-AREA.
00151     001510     03 STACK-KEY.
00152     001520        05 STACK-SUB-KEY.
00153     001530           07 STACK-CODE      PICTURE X       VALUE SPACES.
00154     001540           07 STACK-CATEGORY-1  PICTURE X(38) VALUE SPACES.
00155     001550           07 STACK-AUTHOR-1    PICTURE X(15) VALUE SPACES.
00156     001560          07 STACK-PROGRAM-ID-1 PICTURE X(10) VALUE SPACES.
00157     001570        05 STACK-R-KEY  REDEFINES STACK-SUB-KEY.
00158     001580           07 FILLER             PICTURE X.
00159     001590           07 STACK-AUTHOR-2     PICTURE X(15).
00160     001600           07 STACK-PROGRAM-ID-2 PICTURE X(10).
00161     001610           07 STACK-CATEGORY-2   PICTURE X(38).
00162     001620        05 STACK-DATE            PICTURE X(6) VALUE SPACES.
00163     001630     03 STACK-RUNS               PICTURE S9(5)
00164     001640                                            COMPUTATIONAL-3
00165     001650                                            VALUE +0.
00166     001660     03 STACK-EXCEPTIONS         PICTURE S9(7)
00167     001670                                            COMPUTATIONAL-3
00168     001680                                              VALUE +0.
00169     001690     03 FILLER                   PICTURE X(3) VALUE SPACE.
00170     001700 01  REPORT-HEAD-1.
00171     001710     03 FILLER                 PICTURE X(10)  VALUE SPACES.
00172     001720     03 REPORT-HEAD-DATE       PICTURE XX/XX/XX.
00173     001730     03 FILLER                 PICTURE X(10)  VALUE SPACE.
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 5

05017     003090     02 FILLER        PIC X(49) VALUE
05018     003100     'CA-METACOBOL+ USAGE MANAGEMENT REVIEW REPORT'.
05019     003110     02 FILLER       PIC X(5) VALUE 'PAGE'.
05020     003120     02 REPORT-HEAD-PAGE PIC Z(3).
05021     003130 01  REPORT-HEAD-2.
05022     003140     02 FILLER         PIC X(10) VALUE SPACE.
05023     003150     02 REPORT-ACTIVITY-TEXT PIC X(21).
05024     003160     02 REPORT-ACTIVITY PICTURE X(54).
05025     003170 01  REPORT-HEAD-3.
05026     003180     02 FILLER          PIC X(10) VALUE SPACES.
05027     003190     02 REPORT-HEAD-3-TEXT PIC X(50) VALUE
05028     003200         'AUTHOR          PROGRAM'.
05029     003210     02 FILLER          PIC X(25) VALUE
05030     003220         ' RUNS LAST RUN EXCEPTIONS'.
05031     003280 01  REPORT-DETAIL.
05032     003290     02 FILLER   PIC X(10) VALUE SPACE.
05033     003300     02 REPORT-DETAIL-1.
05034     003310       03 REPORT-AUTHOR-1 PIC X(16).
05035     003320     03 REPORT-PROGRAM-ID-1 PIC X(34).
05036     003330      02 REPORT-DETAIL-2 REDEFINES REPORT-DETAIL-1.
05037     003340       03 REPORT-PROGRAM-ID-2 PIC X(11).
05038     003350       03 REPORT-CATEGORY-2 PIC X(39).
05039     003360     02 REPORT-DETAIL-RUNS PIC Z(5).
05040     003370     02 FILLER          PIC X.
05041     003380     02 REPORT-DETAIL-DATE.
05042     003390       03 REPORT-DETAIL-MO PIC XX.
05043     003400       03 FILLER        PIC X.
05044     003410       03 REPORT-DETAIL-DY PIC XX.
05045     003420       03 FILLER        PIC X.
05046     003430       03 REPORT-DETAIL-YR PIC XX.
05047     003440     02 FILLER          PIC X(4).
05048     003450     02 REPORT-DETAIL-EXCEPTIONS PIC Z(7).
05049     003680 PROCEDURE DIVISION.
05050     003690 SORT-ACCT SECTION 00.
******* NOTE        N99  CQA52A-PROGRAM SEGMENTATION IS COSTLY IN SYSTEM/370 OVERHEAD
05051     003700    SORT TRANS-FILE ASCENDING KEY TRANS-KEY INPUT PROCEDURE
05052     003710       FPOC OUTPUT PROCEDURE LPOC.
******* NOTE        N99  CQA44A-'SORT INPUT PROCEDURE' WITHOUT THRU PHRASE
******* NOTE        N99  CQA44A-'SORT OUTPUT PROCEDURE' WITHOUT THRU PHRASE
05053     003720     GOBACK.
05054     003830 FPOC SECTION 50.
******* NOTE        N99  CQA52A-PROGRAM SEGMENTATION IS COSTLY IN SYSTEM/370 OVERHEAD
05055     003840     OPEN INPUT ACCT-FILE.
05056     003900 FPOC-RECORD-FETCH.
05057     003910     READ ACCT-FILE AT END GO TO FPOC-WINDUP.
******* NOTE        N99  CQA49A-'READ' STATEMENT WITHOUT FROM/INTO OPTION
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05058     003940     IF ACCT-KEY NOT = 'ACCT'
05059     003950     GO TO FPOC-RECORD-FETCH.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05060     003960     ADD +1 TO ACCT-REC-COUNT.
05061     003970     SET X1 X2 X3 TO 1.
05062     003980     MOVE SPACES TO WORK-CATEGORY.
05063     004040 FPOC-FIND-EXCEPTIONS.
05064     004050     IF X1 > 7
05065     004060     GO TO FPOC-FIND-CATEGORY.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05066     004070     IF ACCT-CATEGORY (X1) < '0' OR
05067     004080      ACCT-CATEGORY (X1) > '9'
05068     004090       GO TO FPOC-FIND-CATEGORY.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05069     004100     MOVE ACCT-CATEGORY (X1) TO WORK-BYTE (X2).
05070     004110     SET X1 X2 UP BY 1.
05071     004120     GO TO FPOC-FIND-EXCEPTIONS.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 4    AMR

00174     001740     03 FILLER                  PICTURE X(49)      VALUE
00175     001750                    'CA-METACOBOL+ USAGE MANAGEMENT REVIEW
00176     001760-                               'W REPORT'.
00177     001770     03 FILLER             PICTURE X(5)       VALUE 'PAGE'
00178     001780     03 REPORT-HEAD-PAGE        PICTURE Z(3).
00179     001790 01  REPORT-HEAD-2.
00180     001800     03 FILLER              PICTURE X(10)    VALUE SPACE.
00181     001810     03 REPORT-ACTIVITY-TEXT PICTURE X(21)    VALUE SPACES.
00182     001820     03 REPORT-ACTIVITY     PICTURE X(54)    VALUE SPACES.
00183     001830 01  REPORT-HEAD-3.
00184     001840     03 FILLER              PICTURE X(10)    VALUE SPACES.
00185     001850     03 REPORT-HEAD-3-TEXT      PICTURE X(50)     VALUE
00186     001860                                'AUTHOR          PROGRAM'.
00187     001870     03 FILLER                  PICTURE X(25)      VALUE
00188     001880                             ' RUNS LAST RUN EXCEPTIONS'.
00189     001890 01  REPORT-DETAIL.
00190     001900     03 FILLER             PICTURE X(10)      VALUE SPACE.
00191     001910     03 REPORT-DETAIL-1.
00192     001920        05 REPORT-AUTHOR-1     PICTURE X(16)  VALUE SPACES.
00193     001930        05 REPORT-PROGRAM-ID-1  PICTURE X(34) VALUE SPACES.
00194     001940     03 REPORT-DETAIL-2 REDEFINES REPORT-DETAIL-1.
00195     001950        05 REPORT-PROGRAM-ID-2   PICTURE X(11).
00196     001960        05 REPORT-CATEGORY-2     PICTURE X(39).
00197     001970     03 REPORT-DETAIL-RUNS       PICTURE Z(5).
00198     001980     03 FILLER                PICTURE X      VALUE SPACES.
00199     001990     03 REPORT-DETAIL-DATE.
00200     002000        05 REPORT-DETAIL-MO      PICTURE XX   VALUE SPACES.
00201     002010        05 FILLER                PICTURE X    VALUE SPACES.
00202     002020        05 REPORT-DETAIL-DY      PICTURE XX   VALUE SPACES.
00203     002030        05 FILLER                PICTURE X    VALUE SPACES.
00204     002040        05 REPORT-DETAIL-YR      PICTURE XX   VALUE SPACES.
00205     002050     03 FILLER                  PICTURE X(4) VALUE SPACES.
00206     002060     03 REPORT-DETAIL-EXCEPTIONS PICTURE Z(7).
00207     002070 PROCEDURE DIVISION.
00208     002080 0010-SORT-ACCT SECTION 00.
00209     002090   SORT TRANS-FILE ASCENDING KEY TRANS-KEY INPUT PROCEDURE
00210     002100        0020-FPOC OUTPUT PROCEDURE 0090-LPOC.
00211     002110     GOBACK.
00212     002120 0020-FPOC SECTION 50.
00213     002130     OPEN INPUT ACCT-FILE.
00214     002140 0030-FPOC-RECORD-FETCH.
00215     002150     READ ACCT-FILE
00216     002160        AT END
00217     002170        GO TO 0070-FPOC-WINDUP.
00218     002180     IF ACCT-KEY NOT = 'ACCT'
00219     002190        GO TO 0030-FPOC-RECORD-FETCH.
00220     002200     ADD +1 TO ACCT-REC-COUNT.
00221     002210     SET INDEX-1 INDEX-2 INDEX-3 TO 1.
00222     002220     MOVE SPACES TO WORK-CATEGORY.
00223     002230 0040-FPOC-FIND-EXCEPTIONS.
00224     002240     IF INDEX-1 > 7
00225     002250        GO TO 0050-FPOC-FIND-CATEGORY.
00226     002260     IF ACCT-CATEGORY (INDEX-1) < '0'
00227     002270        OR ACCT-CATEGORY (INDEX-1) > '9'
00228     002280        GO TO 0050-FPOC-FIND-CATEGORY.
00229     002290     MOVE ACCT-CATEGORY (INDEX-1) TO WORK-BYTE (INDEX-2).
00230     002300     SET INDEX-1 INDEX-2 UP BY 1.
00231     002310     GO TO 0040-FPOC-FIND-EXCEPTIONS.
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 6

******* NOTE        N99  CQA55A-SEQUENTIAL TABLE SEARCH
05072     004180 FPOC-FIND-CATEGORY.
05073     004190     IF X1 > 38
05074     004200       GO TO FPOC-RELEASE.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05075     004210     MOVE ACCT-CATEGORY (X1) TO WORK-CHARACTER (X3).
05076     004220     SET X1 X3 UP BY 1.
05077     004230     GO TO FPOC-FIND-CATEGORY.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
******* NOTE        N99  CQA55A-SEQUENTIAL TABLE SEARCH
05078     004240     EJECT
05079     004320 FPOC-RELEASE.
05080     004330     IF WORK-BYTE (7) = ' '
05081     004340       MOVE WORK-BYTE (6) TO WORK-BYTE (7)
05082     004350       MOVE WORK-BYTE (5) TO WORK-BYTE (6)
05083     004360       MOVE WORK-BYTE (4) TO WORK-BYTE (5)
05084     004370       MOVE WORK-BYTE (3) TO WORK-BYTE (4)
05085     004380       MOVE WORK-BYTE (2) TO WORK-BYTE (3)
05086     004390       MOVE WORK-BYTE (1) TO WORK-BYTE (2)
05087     004400     MOVE '0' TO WORK-BYTE (1)
05088     004410     GO TO FPOC-RELEASE.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
******* NOTE        N99  CQA55A-SEQUENTIAL TABLE SEARCH
05089     004420     MOVE ACCT-YR TO WORK-YR.
05090     004430     MOVE ACCT-MO TO WORK-MO.
05091     004440     MOVE ACCT-DY TO WORK-DY.
05092     004450     MOVE WORK-DATE TO SORT-DATE.
05093     004460     MOVE ACCT-REC-COUNT TO SORT-SEQ.
05094     004470     MOVE +1 TO SORT-RUNS.
05095     004480     MOVE WORK-NUMERIC TO SORT-EXCEPTIONS.
05096     004490     MOVE '1' TO SORT-CODE.
05097     004500     MOVE WORK-CATEGORY TO SORT-CATEGORY-1.
05098     004510     MOVE ACCT-AUTHOR TO SORT-AUTHOR-1.
05099     004520     MOVE ACCT-PROGRAM-ID TO SORT-PROGRAM-ID-1.
05100     004530     RELEASE TRANS-RECORD FROM SORT-RECORD.
******* NOTE        N99  CQA48A-'RELEASE' STATEMENT WITH 'FROM' OPTION
05101     004540     MOVE '2' TO SORT-CODE.
05102     004550     MOVE WORK-CATEGORY TO SORT-CATEGORY-2.
05103     004560     MOVE ACCT-AUTHOR TO SORT-AUTHOR-2.
05104     004570     MOVE ACCT-PROGRAM-ID TO SORT-PROGRAM-ID-2.
05105     004580     RELEASE TRANS-RECORD FROM SORT-RECORD.
******* NOTE        N99  CQA48A-'RELEASE' STATEMENT WITH 'FROM' OPTION
05106     004590     GO TO FPOC-RECORD-FETCH.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05107     004650 FPOC-WINDUP.
05108     004660     CLOSE ACCT-FILE.
05109     004670 FPOC-EXIT.
05110     004680     EXIT.
05111     004790 LPOC SECTION 60.
******* NOTE        N99  CQA52A-PROGRAM SEGMENTATION IS COSTLY IN SYSTEM/370 OVERHEAD
05112     004800     OPEN INPUT OLD-FILE OUTPUT NEW-FILE REPORT-FILE.
05113     004810     ACCEPT ACCEPT-DATE FROM DATE.
******* NOTE        N99  CQA31A-'ACCEPT-DATE' AND 'DATE'
******* NOTE        N99         HAVE DIFFERING USAGES IN 'ACCEPT' STATEMENT
05114     004820     MOVE A-YEAR TO D-YEAR
05115     004830     MOVE A-MONTH TO D-MONTH
05116     004840     MOVE A-DAY TO D-DAY
05117     004850     MOVE AMERICAN-DATE-FORMAT TO REPORT-HEAD-DATE.
05118     004860     PERFORM LPOC-READ-OLD.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05119     004870     RETURN TRANS-FILE INTO SORT-RECORD
******* NOTE        N99  CQA48A-'RETURN' STATEMENT WITH 'INTO' OPTION
05120     004880       AT END
05121     004890       MOVE HIGH-VALUE TO SORT-KEY.
05122     004900     PERFORM LPOC-STACK THRU LPOC-STACK-EXIT.
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 5    AMR

00232     002320 0050-FPOC-FIND-CATEGORY.
00233     002330     IF INDEX-1 > 38
00234     002340        GO TO 0060-FPOC-RELEASE.
00235     002350  MOVE ACCT-CATEGORY (INDEX-1) TO WORK-CHARACTER (INDEX-3).
00236     002360     SET INDEX-1 INDEX-3 UP BY 1.
00237     002370     GO TO 0050-FPOC-FIND-CATEGORY.
          002380     EJECT
00238     002390 0060-FPOC-RELEASE.
00239     002400     IF WORK-BYTE (7) = ' '
00240     002410        MOVE WORK-BYTE (6) TO WORK-BYTE (7)
00241     002420        MOVE WORK-BYTE (5) TO WORK-BYTE (6)
00242     002430        MOVE WORK-BYTE (4) TO WORK-BYTE (5)
00243     002440        MOVE WORK-BYTE (3) TO WORK-BYTE (4)
00244     002450        MOVE WORK-BYTE (2) TO WORK-BYTE (3)
00245     002460        MOVE WORK-BYTE (1) TO WORK-BYTE (2)
00246     002470        MOVE '0' TO WORK-BYTE (1)
00247     002480        GO TO 0060-FPOC-RELEASE.
00248     002490     MOVE ACCT-YR TO WORK-YR.
00249     002500     MOVE ACCT-MO TO WORK-MO.
00250     002510     MOVE ACCT-DY TO WORK-DY.
00251     002520     MOVE WORK-DATE TO SORT-DATE.
00252     002530     MOVE ACCT-REC-COUNT TO SORT-SEQ.
00253     002540     MOVE +1 TO SORT-RUNS.
00254     002550     MOVE WORK-NUMERIC TO SORT-EXCEPTIONS.
00255     002560     MOVE '1' TO SORT-CODE.
00256     002570     MOVE WORK-CATEGORY TO SORT-CATEGORY-1.
00257     002580     MOVE ACCT-AUTHOR TO SORT-AUTHOR-1.
00258     002590     MOVE ACCT-PROGRAM-ID TO SORT-PROGRAM-ID-1.
00259     002600     RELEASE TRANS-RECORD FROM SORT-RECORD.
00260     002610     MOVE '2' TO SORT-CODE.
00261     002620     MOVE WORK-CATEGORY TO SORT-CATEGORY-2.
00262     002630     MOVE ACCT-AUTHOR TO SORT-AUTHOR-2.
00263     002640     MOVE ACCT-PROGRAM-ID TO SORT-PROGRAM-ID-2.
00264     002650     RELEASE TRANS-RECORD FROM SORT-RECORD.
00265     002660     GO TO 0030-FPOC-RECORD-FETCH.
00266     002670 0070-FPOC-WINDUP.
00267     002680     CLOSE ACCT-FILE.
00268     002690 0080-FPOC-EXIT.
00269     002700     EXIT.
00270     002710 0090-LPOC SECTION 60.
00271     002720     OPEN INPUT OLD-FILE OUTPUT NEW-FILE REPORT-FILE.
00272     002730     ACCEPT ACCEPT-DATE FROM DATE.
00273     002740     MOVE ACCEPT-YEAR TO DECODED-YEAR
00274     002750     MOVE ACCEPT-MONTH TO DECODED-MONTH
00275     002760     MOVE ACCEPT-DAY TO DECODED-DAY
00276     002770     MOVE AMERICAN-DATE-FORMAT TO REPORT-HEAD-DATE.
00277     002780     PERFORM 0110-LPOC-READ-OLD.
00278     002790     RETURN TRANS-FILE INTO SORT-RECORD
00279     002800        AT END
00280     002810        MOVE HIGH-VALUE TO SORT-KEY.
00281     002820     PERFORM 0120-LPOC-STACK THRU 0140-LPOC-STACK-EXIT.
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 7

******* NOTE        N99  CQA45A-'PERFORM' WITH THRU PHRASE
05123     004990 LPOC-MATCH.
05124     005000     IF OLD-SUB-KEY < STACK-SUB-KEY
******* NOTE        N99  CQA30A-'STACK-SUB-KEY' IS A GROUP ITEM
******* NOTE        N99         IN CONDITIONAL STATEMENT
05125     005010       MOVE OLD-WORK TO NEW-RECORD
05126     005020       PERFORM LPOC-OUTPUT THRU LPOC-OUTPUT-EXIT
05127     005030       PERFORM LPOC-READ-OLD
******* NOTE        N99  CQA45A-'PERFORM' WITH THRU PHRASE
05128     005040       GO TO LPOC-MATCH.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05129     005050     IF OLD-SUB-KEY > STACK-SUB-KEY
******* NOTE        N99  CQA30A-'STACK-SUB-KEY' IS A GROUP ITEM
******* NOTE        N99         IN CONDITIONAL STATEMENT
05130     005060       MOVE STACK-AREA TO NEW-RECORD
05131     005070       PERFORM LPOC-OUTPUT THRU LPOC-OUTPUT-EXIT
05132     005080       PERFORM LPOC-STACK THRU LPOC-STACK-EXIT
******* NOTE        N99  CQA45A-'PERFORM' WITH THRU PHRASE
05133     005090       GO TO LPOC-MATCH.
******* NOTE        N99  CQA45A-'PERFORM' WITH THRU PHRASE
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05134     005100     IF OLD-KEY = HIGH-VALUE
05135     005110       GO TO LPOC-WINDUP.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05136     005120     MOVE OLD-WORK TO NEW-RECORD.
05137     005130     IF STACK-DATE NOT < NEW-DATE
******* NOTE        N99  CQA30A-'NEW-DATE' IS A GROUP ITEM
******* NOTE        N99         IN CONDITIONAL STATEMENT
05138     005140       MOVE STACK-DATE TO NEW-DATE
05139     005150       MOVE STACK-EXCEPTIONS TO NEW-EXCEPTIONS
05140     005160       ADD STACK-RUNS TO NEW-RUNS.
05141     005170     PERFORM LPOC-OUTPUT THRU LPOC-OUTPUT-EXIT.
******* NOTE        N99  CQA45A-'PERFORM' WITH THRU PHRASE
05142     005180     PERFORM LPOC-READ-OLD.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05143     005190     PERFORM LPOC-STACK THRU LPOC-STACK-EXIT.
******* NOTE        N99  CQA45A-'PERFORM' WITH THRU PHRASE
05144     005200     GO TO LPOC-MATCH.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05145     005270 LPOC-READ-OLD.
05146     005280     READ OLD-FILE INTO OLD-WORK
******* NOTE        N99  CQA48A-'READ' STATEMENT WITH 'INTO' OPTION
05147     005290       AT END
05148     005300       MOVE HIGH-VALUE TO OLD-KEY.
05149     005390 LPOC-STACK.
05150     005400     IF SORT-KEY = HIGH-VALUE
05151     005410       MOVE HIGH-VALUE TO STACK-KEY
05152     005420       MOVE ZERO TO STACK-RUNS STACK-EXCEPTIONS
05153     005430       GO TO LPOC-STACK-EXIT.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05154     005440     MOVE SORT-KEY TO STACK-KEY.
05155     005450     MOVE SORT-RUNS TO STACK-RUNS.
05156     005460     MOVE SORT-EXCEPTIONS TO STACK-EXCEPTIONS.
05157     005470 LPOC-STACK-RETURN.
05158     005480     RETURN TRANS-FILE INTO SORT-RECORD
******* NOTE        N99  CQA48A-'RETURN' STATEMENT WITH 'INTO' OPTION
05159     005490       AT END
05160     005500       MOVE HIGH-VALUE TO SORT-KEY
05161     005510     GO TO LPOC-STACK-EXIT.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05162     005520     IF SORT-SUB-KEY = STACK-SUB-KEY
******* NOTE        N99  CQA30A-'STACK-SUB-KEY' IS A GROUP ITEM
******* NOTE        N99         IN CONDITIONAL STATEMENT
******* NOTE        N99  CQA30A-'SORT-SUB-KEY' IS A GROUP ITEM
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 6    AMR

00282     002830 0100-LPOC-MATCH.
00283     002840     IF OLD-SUB-KEY < STACK-SUB-KEY
00284     002850        MOVE OLD-WORK TO NEW-RECORD
00285     002860        PERFORM 0150-LPOC-OUTPUT THRU
0190-LPOC-OUTPUT-EXIT
00286     002870        PERFORM 0110-LPOC-READ-OLD
00287     002880        GO TO 0100-LPOC-MATCH.
00288     002890     IF OLD-SUB-KEY > STACK-SUB-KEY
00289     002900        MOVE STACK-AREA TO NEW-RECORD
00290     002910        PERFORM 0150-LPOC-OUTPUT THRU
0190-LPOC-OUTPUT-EXIT
00291     002920        PERFORM 0120-LPOC-STACK THRU
0140-LPOC-STACK-EXIT
00292     002930        GO TO 0100-LPOC-MATCH.
00293     002940     IF OLD-KEY = HIGH-VALUE
00294     002950        GO TO 0230-LPOC-WINDUP.
00295     002960     MOVE OLD-WORK TO NEW-RECORD.
00296     002970     IF STACK-DATE NOT < NEW-DATE
00297     002980        MOVE STACK-DATE TO NEW-DATE
00298     002990        MOVE STACK-EXCEPTIONS TO NEW-EXCEPTIONS
00299     003000        ADD STACK-RUNS TO NEW-RUNS.
00300     003010     PERFORM 0150-LPOC-OUTPUT THRU
0190-LPOC-OUTPUT-EXIT.
00301     003020     PERFORM 0110-LPOC-READ-OLD.
00302     003030     PERFORM 0120-LPOC-STACK THRU
0140-LPOC-STACK-EXIT.
00303     003040     GO TO 0100-LPOC-MATCH.
00304     003050 0110-LPOC-READ-OLD.
00305     003060     READ OLD-FILE INTO OLD-WORK
00306     003070        AT END
00307     003080        MOVE HIGH-VALUE TO OLD-KEY.
00308     003090 0120-LPOC-STACK.
00309     003100     IF SORT-KEY = HIGH-VALUE
00310     003110        MOVE HIGH-VALUE TO STACK-KEY
00311     003120        MOVE ZERO TO STACK-RUNS STACK-EXCEPTIONS
00312     003130        GO TO 0140-LPOC-STACK-EXIT.
00313     003140     MOVE SORT-KEY TO STACK-KEY.
00314     003150     MOVE SORT-RUNS TO STACK-RUNS.
00315     003160     MOVE SORT-EXCEPTIONS TO STACK-EXCEPTIONS.
00316     003170 0130-LPOC-STACK-RETURN.
00317     003180     RETURN TRANS-FILE INTO SORT-RECORD
00318     003190        AT END
00319     003200        MOVE HIGH-VALUE TO SORT-KEY
00320     003210        GO TO 0140-LPOC-STACK-EXIT.
00321     003220     IF SORT-SUB-KEY = STACK-SUB-KEY
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 8

******* NOTE        N99         IN CONDITIONAL STATEMENT
05163     005530       MOVE SORT-DATE TO STACK-DATE
05164     005540       MOVE SORT-EXCEPTIONS TO STACK-EXCEPTIONS
05165     005550       ADD SORT-RUNS TO STACK-RUNS
05166     005560     GO TO LPOC-STACK-RETURN.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05167     005570 LPOC-STACK-EXIT.
05168     005580     EXIT.
05169     005670 LPOC-OUTPUT.
05170     005680     IF NEW-CODE NOT = REPORT-CODE
05171     005690     PERFORM LPOC-RESET.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05172     005700     MOVE SPACES TO REPORT-DETAIL FORM-CONTROL.
05173     005710     IF REPORT-CODE = '2'
05174     005720     GO TO LPOC-OUTPUT-2.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05175     005780 LPOC-OUTPUT-1.
05176     005790     IF NEW-CATEGORY-1 NOT = PREV-CATEGORY
05177     005800       MOVE +57 TO DEPTH-COUNTER
05178     005810       MOVE NEW-CATEGORY-1 TO REPORT-ACTIVITY PREV-CATEGORY.
05179     005820     ADD +1 TO DEPTH-COUNTER.
05180     005830     IF NEW-AUTHOR-1 NOT = PREV-AUTHOR
05181     005840       ADD +1 TO DEPTH-COUNTER
05182     005850       MOVE '0' TO FORM-CONTROL
05183     005860       MOVE NEW-AUTHOR-1 TO PREV-AUTHOR REPORT-AUTHOR-1.
05184     005870     IF DEPTH-COUNTER > +56
05185     005880       MOVE PREV-AUTHOR TO REPORT-AUTHOR-1
05186     005890       PERFORM LPOC-REPORT-HEAD.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05187     005900     MOVE NEW-PROGRAM-ID-1 TO REPORT-PROGRAM-ID-1.
05188     005910     GO TO LPOC-OUTPUT-COMMON.
******* NOTE        N99  CQA39A-'GO' CAN BE DANGEROUS
05189     005970 LPOC-OUTPUT-2.
05190     005980     IF NEW-AUTHOR-2 NOT = PREV-AUTHOR
05191     005990       MOVE +57 TO DEPTH-COUNTER
05192     006000       MOVE NEW-AUTHOR-2 TO REPORT-ACTIVITY PREV-AUTHOR.
05193     006010     ADD +1 TO DEPTH-COUNTER.
05194     006020     IF NEW-PROGRAM-ID-2 NOT = PREV-PROGRAM-ID
05195     006030       ADD +1 TO DEPTH-COUNTER
05196     006040       MOVE '0' TO FORM-CONTROL
05197     006050       MOVE NEW-PROGRAM-ID-2 TO PREV-PROGRAM-ID
05198     006060         REPORT-PROGRAM-ID-2.
05199     006070     IF DEPTH-COUNTER > +56
05200     006080       MOVE PREV-PROGRAM-ID TO REPORT-PROGRAM-ID-2
05201     006090       PERFORM LPOC-REPORT-HEAD.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05202     006100     MOVE NEW-CATEGORY-2 TO REPORT-CATEGORY-2.
05203     006170 LPOC-OUTPUT-COMMON.
05204     006180     MOVE NEW-RUNS TO REPORT-DETAIL-RUNS.
05205     006190     MOVE '  /  /  ' TO REPORT-DETAIL-DATE.
05206     006200     MOVE NEW-YR TO REPORT-DETAIL-YR.
05207     006210     MOVE NEW-MO TO REPORT-DETAIL-MO.
05208     006220     MOVE NEW-DY TO REPORT-DETAIL-DY.
05209     006230     MOVE NEW-EXCEPTIONS TO REPORT-DETAIL-EXCEPTIONS.
05210     006240     MOVE REPORT-DETAIL TO REPORT-RECORD.
05211     006250     PERFORM LPOC-PRINT.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05212     006260     MOVE SPACE TO FORM-CONTROL.
05213     006270     WRITE NEW-RECORD.
******* NOTE        N99  CQA49A-'WRITE' STATEMENT WITHOUT FROM/INTO OPTION
05214     006280 LPOC-OUTPUT-EXIT.
05215     006290     EXIT.
05216     006370 LPOC-RESET.
05217     006380     MOVE NEW-CODE TO REPORT-CODE.
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 7    AMR

00322     003230        MOVE SORT-DATE TO STACK-DATE
00323     003240        MOVE SORT-EXCEPTIONS TO STACK-EXCEPTIONS
00324     003250        ADD SORT-RUNS TO STACK-RUNS
00325     003260        GO TO 0130-LPOC-STACK-RETURN.
00326     003270 0140-LPOC-STACK-EXIT.
00327     003280     EXIT.
00328     003290 0150-LPOC-OUTPUT.
00329     003300     IF NEW-CODE NOT = REPORT-CODE
00330     003310        PERFORM 0200-LPOC-RESET.
00331     003320     MOVE SPACES TO REPORT-DETAIL FORM-CONTROL.
00332     003330     IF REPORT-CODE = '2'
00333     003340        GO TO 0170-LPOC-OUTPUT-2.
00334     003350 0160-LPOC-OUTPUT-1.
00335     003360     IF NEW-CATEGORY-1 NOT = PREVIOUS-CATEGORY
00336     003370        MOVE +57 TO DEPTH-COUNTER
00337     003380        MOVE NEW-CATEGORY-1 TO REPORT-ACTIVITY
PREVIOUS-CATEGORY
00338     003390     ADD +1 TO DEPTH-COUNTER.
00339     003400     IF NEW-AUTHOR-1 NOT = PREVIOUS-AUTHOR
00340     003410        ADD +1 TO DEPTH-COUNTER
00341     003420        MOVE '0' TO FORM-CONTROL
00342     003430        MOVE NEW-AUTHOR-1 TO PREVIOUS-AUTHOR REPORT-AUTHOR-1.
00343     003440     IF DEPTH-COUNTER > +56
00344     003450        MOVE PREVIOUS-AUTHOR TO REPORT-AUTHOR-1
00345     003460        PERFORM 0210-LPOC-REPORT-HEAD.
00346     003470     MOVE NEW-PROGRAM-ID-1 TO REPORT-PROGRAM-ID-1.
00347     003480     GO TO 0180-LPOC-OUTPUT-COMMON.
00348     003490 0170-LPOC-OUTPUT-2.
00349     003500     IF NEW-AUTHOR-2 NOT = PREVIOUS-AUTHOR
00350     003510        MOVE +57 TO DEPTH-COUNTER
00351     003520        MOVE NEW-AUTHOR-2 TO REPORT-ACTIVITY PREVIOUS-AUTHOR.
00352     003530     ADD +1 TO DEPTH-COUNTER.
00353     003540     IF NEW-PROGRAM-ID-2 NOT = PREVIOUS-PROGRAM-ID
00354     003550        ADD +1 TO DEPTH-COUNTER
00355     003560        MOVE '0' TO FORM-CONTROL
00356     003570        MOVE NEW-PROGRAM-ID-2 TO PREVIOUS-PROGRAM-ID
00357     003580           REPORT-PROGRAM-ID-2.
00358     003590     IF DEPTH-COUNTER > +56
00359     003600        MOVE PREVIOUS-PROGRAM-ID TO REPORT-PROGRAM-ID-2
00360     003610        PERFORM 0210-LPOC-REPORT-HEAD.
00361     003620     MOVE NEW-CATEGORY-2 TO REPORT-CATEGORY-2.
00362     003630 0180-LPOC-OUTPUT-COMMON.
00363     003640     MOVE NEW-RUNS TO REPORT-DETAIL-RUNS.
00364     003650     MOVE '  /  /  ' TO REPORT-DETAIL-DATE.
00365     003660     MOVE NEW-YR TO REPORT-DETAIL-YR.
00366     003670     MOVE NEW-MO TO REPORT-DETAIL-MO.
00367     003680     MOVE NEW-DY TO REPORT-DETAIL-DY.
00368     003690     MOVE NEW-EXCEPTIONS TO REPORT-DETAIL-EXCEPTIONS.
00369     003700     MOVE REPORT-DETAIL TO REPORT-RECORD.
00370     003710     PERFORM 0220-LPOC-PRINT.
00371     003720     MOVE SPACE TO FORM-CONTROL.
00372     003730     WRITE NEW-RECORD.
00373     003740 0190-LPOC-OUTPUT-EXIT.
00374     003750     EXIT.
00375     003760 0200-LPOC-RESET.
00376     003770     MOVE NEW-CODE TO REPORT-CODE.
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 9

05218     006390     IF REPORT-CODE = '1' MOVE
05219     006400       'ACTIVITY BY CATEGORY' TO REPORT-ACTIVITY-TEXT
05220     006410        ELSE MOVE
05221     006420       'ACTIVITY BY AUTHOR' TO REPORT-ACTIVITY-TEXT
05222     006430     MOVE 'PROGRAM    CATEGORY' TO REPORT-HEAD-3-TEXT.
05223     006440     MOVE +57 TO DEPTH-COUNTER.
05224     006450     MOVE +0 TO REPORT-PAGE.
05225     006460     MOVE LOW-VALUE TO PREV-CATEGORY PREV-AUTHOR
05226     006520 LPOC-REPORT-HEAD.
05227     006530     ADD +1 TO REPORT-PAGE.
05228     006540     MOVE REPORT-PAGE TO REPORT-HEAD-PAGE.
05229     006550     MOVE REPORT-HEAD-1 TO REPORT-RECORD.
05230     006560     MOVE '1' TO FORM-CONTROL.
05231     006570     PERFORM LPOC-PRINT.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05232     006580     MOVE REPORT-HEAD-2 TO REPORT-RECORD.
05233     006590     MOVE SPACE TO FORM-CONTROL.
05234     006600     PERFORM LPOC-PRINT.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05235     006610     MOVE REPORT-HEAD-3 TO REPORT-RECORD.
05236     006620     MOVE '0' TO FORM-CONTROL.
05237     006630     PERFORM LPOC-PRINT.
******* NOTE        N99  CQA44A-'PERFORM' WITHOUT THRU PHRASE
05238     006640     MOVE +1 TO DEPTH-COUNTER.
05239     006650     MOVE '-' TO FORM-CONTROL.
05240     006710 LPOC-PRINT.
05241     006720     IF FORM-CONTROL = ' ' WRITE REPORT-RECORD AFTER
05242     006730       1 LINE ELSE IF FORM-CONTROL = '0'
******* NOTE        N99  CQA49A-'WRITE' STATEMENT WITHOUT FROM/INTO OPTION
05243     006760       WRITE REPORT-RECORD AFTER ADVANCING 2 LINES ELSE
******* NOTE        N99  CQA49A-'WRITE' STATEMENT WITHOUT FROM/INTO OPTION
05244     006780       IF FORM-CONTROL = '-'
05245     006790       WRITE REPORT-RECORD AFTER ADVANCING 3 LINES ELSE
******* NOTE        N99  CQA49A-'WRITE' STATEMENT WITHOUT FROM/INTO OPTION
05246     006810       IF FORM-CONTROL = '1'
05247     006820       WRITE REPORT-RECORD AFTER ADVANCING TOP-OF-PAGE.
******* NOTE        N99  CQA49A-'WRITE' STATEMENT WITHOUT FROM/INTO OPTION
05248     006880 LPOC-WINDUP.
05249     006890     CLOSE NEW-FILE OLD-FILE REPORT-FILE.
05250     006900     OPEN OUTPUT ACCT-FILE.
05251     006901     MOVE SPACES TO ACCT-RECORD.
05252     006902     WRITE ACCT-RECORD.
******* NOTE        N99  CQA49A-'WRITE' STATEMENT WITHOUT FROM/INTO OPTION
05253     006903     CLOSE ACCT-FILE.
05254     006910 LPOC-EXIT.
05255     006920     EXIT.
******* NOTE        N99  * - - - - - - - - - - - - - - - - - - - - - - *
******* NOTE        N99   ALL INPUT SOURCE CODE HAS BEEN PROCESSED.
******* NOTE        N99   THE DIAGNOSTICS WHICH FOLLOW, REFER TO THE
******* NOTE        N99   PRECEDING INPUT SOURCE CODE.
******* NOTE        N99  * - - - - - - - - - - - - - - - - - - - - - - *
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99         LINE NUMBER = 005051, SEQ NUM = 003700
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99         LINE NUMBER = 005051, SEQ NUM = 003700
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99         LINE NUMBER = 005057, SEQ NUM = 003910
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99         LINE NUMBER = 005059, SEQ NUM = 003950
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99         LINE NUMBER = 005065, SEQ NUM = 004060
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99         LINE NUMBER = 005068, SEQ NUM = 004090
```

Output Listing

```
CA-METACOBOL+ OUTPUT     V1.0   01/04/92 13:04  PAGE 8    AMR

00377     003780     IF REPORT-CODE = '1'
00378     003790        MOVE 'ACTIVITY BY CATEGORY' TO
REPORT-ACTIVITY-TEXT
00379     003800     ELSE
00380     003810        MOVE 'ACTIVITY BY AUTHOR' TO
REPORT-ACTIVITY-TEXT
00381     003820        MOVE 'PROGRAM    CATEGORY' TO
REPORT-HEAD-3-TEXT.
00382     003830     MOVE +57 TO DEPTH-COUNTER.
00383     003840     MOVE +0 TO REPORT-PAGE.
00384     003850     MOVE LOW-VALUE TO PREVIOUS-CATEGORY
PREVIOUS-AUTHOR
00385     003860        PREVIOUS-PROGRAM-ID.
00386     003870 0210-LPOC-REPORT-HEAD.
00387     003880     ADD +1 TO REPORT-PAGE.
00388     003890     MOVE REPORT-PAGE TO REPORT-HEAD-PAGE.
00389     003900     MOVE REPORT-HEAD-1 TO REPORT-RECORD.
00390     003910     MOVE '1' TO FORM-CONTROL.
00391     003920     PERFORM 0220-LPOC-PRINT.
00392     003930     MOVE REPORT-HEAD-2 TO REPORT-RECORD.
00393     003940     MOVE SPACE TO FORM-CONTROL.
00394     003950     PERFORM 0220-LPOC-PRINT.
00395     003960     MOVE REPORT-HEAD-3 TO REPORT-RECORD.
00396     003970     MOVE '0' TO FORM-CONTROL.
00397     003980     PERFORM 0220-LPOC-PRINT.
00398     003990     MOVE +1 TO DEPTH-COUNTER.
00399     004000     MOVE '-' TO FORM-CONTROL.
00400     004010 0220-LPOC-PRINT.
00401     004020     IF FORM-CONTROL = ' '
00402     004030        WRITE REPORT-RECORD AFTER ADVANCING 1 LINE
00403     004040     ELSE
00404     004050        IF FORM-CONTROL = '0'
00405     004060           WRITE REPORT-RECORD AFTER ADVANCING 2 LINES
00406     004070        ELSE
00407     004080           IF FORM-CONTROL = '-'
00408     004090              WRITE REPORT-RECORD AFTER ADVANCING 3
LINES
00409     004100           ELSE
00410     004110              IF FORM-CONTROL = '1'
00411     004120            WRITE REPORT-RECORD AFTER ADVANCING
TOP-OF-PAGE
00412     004130 0230-LPOC-WINDUP.
00413     004140     CLOSE NEW-FILE OLD-FILE REPORT-FILE.
00414     004150     OPEN OUTPUT ACCT-FILE.
00415     004160     MOVE SPACES TO ACCT-RECORD.
00416     004170     WRITE ACCT-RECORD.
00417     004180     CLOSE ACCT-FILE.
00418     004190 0240-LPOC-EXIT.
00419     004200     EXIT.
```

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 10

******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005071, SEQUENCE NUMBER = 004120
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005074, SEQUENCE NUMBER = 004200
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005077, SEQUENCE NUMBER = 004230
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005088, SEQUENCE NUMBER = 004410
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005106, SEQUENCE NUMBER = 004590
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005118, SEQUENCE NUMBER = 004860
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005122, SEQUENCE NUMBER = 004900
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005122, SEQUENCE NUMBER = 004900
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005126, SEQUENCE NUMBER = 005020
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005126, SEQUENCE NUMBER = 005020
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005127, SEQUENCE NUMBER = 005030
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005128, SEQUENCE NUMBER = 005040
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005131, SEQUENCE NUMBER = 005070
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005131, SEQUENCE NUMBER = 005070
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005132, SEQUENCE NUMBER = 005080
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005132, SEQUENCE NUMBER = 005080
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005133, SEQUENCE NUMBER = 005090
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005135, SEQUENCE NUMBER = 005110
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005141, SEQUENCE NUMBER = 005170
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005141, SEQUENCE NUMBER = 005170
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005142, SEQUENCE NUMBER = 005180
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005143, SEQUENCE NUMBER = 005190
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005143, SEQUENCE NUMBER = 005190
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005144, SEQUENCE NUMBER = 005200
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005153, SEQUENCE NUMBER = 005430
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005161, SEQUENCE NUMBER = 005510
******* NOTE        N99  CQA41A-BACKWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005166, SEQUENCE NUMBER = 005560
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005171, SEQUENCE NUMBER = 005690
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005174, SEQUENCE NUMBER = 005720
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005186, SEQUENCE NUMBER = 005890
******* NOTE        N99  CQA40A-FORWARD 'GO TO'
******* NOTE        N99  LINE NUMBER = 005188, SEQUENCE NUMBER = 005910
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005201, SEQUENCE NUMBER = 006090
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
```

Output Listing

This page is blank by intention, because no output is produced by the input.

Input Listing

```
CA-METACOBOL+ INPUT      V1.0   01/04/92 13:04  PAGE 11

******* NOTE        N99  LINE NUMBER = 005211, SEQUENCE NUMBER = 006250
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005231, SEQUENCE NUMBER = 006570
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005234, SEQUENCE NUMBER = 006600
******* NOTE        N99  CQA46A-FORWARD PROCEDURE INVOCATION
******* NOTE        N99  LINE NUMBER = 005237, SEQUENCE NUMBER = 006630
```

Output (Auxiliary) Listing

```
CA-METACOBOL+ AUXILIARY V1.0   01/04/92 13:04  PAGE 9   AMR

          */
          ** CQA DIAGNOSTIC SUMMARY
          **
          **  CQA30A-GROUP ITEMS USED IN COMPARISONS              5
          **  CQA31A-INCONSISTENT DATA-ITEM USAGES                1
          **  CQA39A-'GO TO' VIOLATIONS                           18
          **  CQA40A-FORWARD BRANCHES (GO TO)                     9
          **  CQA41A-BACKWARD BRANCHES (GO TO)                    9
          **  CQA44A-'PERFORM', 'SORT' OR 'MERGE' WITHOUT 'THRU'  12
          **  CQA45A-'PERFORM', 'SORT' OR 'MERGE' WITH 'THRU'     6
          **  CQA46A-FORWARD PROCEDURE INVOCATION                 24
          **  CQA48A-I/O VERB WITH 'FROM/INTO' OPTION             5
          **  CQA49A-I/O VERB WITHOUT 'FROM/INTO' OPTION          7
          **  CQA52A-SEGMENTED SECTIONS                           3
          **  CQA55A-POSSIBLE SEQUENTIAL TABLE SEARCHES           3
```

Appendix B. Diagnostics

Diagnostics described in this appendix are NOTE or internal error diagnostics produced by either CQA, DCQA, SQA, DSQA, VQA, or DVQA. This section provides an explanation for each NOTE diagnostic produced by CQA. Corrective action is provided for error and warning diagnostics. Corrective action for advisory diagnostics is user-defined unless otherwise indicated.

ERRMSG and VERBOSE Listings
To obtain a listing of all possible error messages, create a COBOL program with the program ID **ERRMSG** and run the program through the CA-MetaCOBOL translator, using the appropriate macro set. For example, for CQA diagnostics, use the CQA Macro Set. To obtain a more detailed listing of all possible error messages, create a COBOL program with the program ID **VERBOSE** and run the program through the CA-MetaCOBOL translator, using the appropriate macro set.

B.1 CQA/DCQA NOTE Diagnostics

NOTE diagnostics contained within CQA begin with the following prefix:
Format:

CQA*nnz*

CQA represents the CQA diagnostic abbreviation.

nn represents the diagnostic number unique within CQA.

z represents one of the following severity codes:

E an *error* requiring correction and retranslation; associated with a condition code of 12.

W a *warning*, possibly requiring correction and retranslation; associated with a condition code of 8.

A an *advisory* comment; associated with a condition code of 4.

B.1.1 Standard CQA/DCQA Messages

CQA01A THIS 'SAME AREA' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE
Explanation: The SAME AREA clause places several files within the same buffer in order to save storage. As a result, an overlay of records may occur which will produce data not expected by the program.

Action: Check program results.

CQA02E TOO MANY NESTED SEARCH STATEMENTS
Explanation: More than 100 SEARCH statements have been nested.

Action: Respecify values for the variables that support this check, or follow your site standards.

CQA03A THIS 'GO TO ... DEPENDING ON ...' STATEMENT MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE
Explanation: The GO TO . . . DEPENDING ON . . . statement may cause serious problems during program maintenance.

Action: Be careful with the control variable, and document program flow control for future reference.

CQA04A THIS LONG RELATION '*long statement*' HAS BEEN CHANGED TO A SHORT RELATION AS FOLLOWS: '*short statement*'
Explanation: The $CQA-RELATIONS SHORT option standardizes all relational operators to a short form, for example, IS EQUAL TO becomes =. IF, PERFORM, SEARCH, and START conditions are standardized. For DCQA, the DL WHERE statement is also standardized.

Action: No action is required.

CQA05A THIS SHORT RELATION '*short statement*' HAS BEEN CHANGED TO A LONG RELATION AS FOLLOWS: '*long statement*'
Explanation: The $CQA-RELATIONS LONG option standardizes all relational operators to a long form, for example, = becomes IS EQUAL TO. IF, PERFORM, SEARCH, and START conditions are standardized. For DCQA, the DL WHERE statement is also standardized.

Action: No action is required.

CQA06A THIS ELEMENTARY ITEM HAS A 'VALUE' '*dataname*' WHICH CONFLICTS WITH A GROUP ITEM THAT ALSO HAS A 'VALUE' CLAUSE
Explanation: An elementary item with a VALUE clause is subordinate to a group level item that also has a VALUE clause.

Action: Remove this conflict.

CQA07A THIS 'BLOCK CONTAINS' CLAUSE IS NOT NEEDED. '*filename*' IS A VSAM FILE
Explanation: The BLOCK CONTAINS clause is ignored when the file is a VSAM file.

Action: No action is required. However, you can remove the BLOCK CONTAINS clause.

CQA09E INVALID $TACK FUNCTION CODE
Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL Support.

CQA10E THE GENERATED LEVEL NUMBER '*number*' FOR '*dataname*' IS INCORRECT. THE $CQA-INCR PARAMETER IS TOO LARGE
Explanation: A level number greater than 49 was generated during Data Division level number normalization.

Action: Specify a smaller increment for the $CQA-INCR control statement.

CQA11W THE COPYBOOK '*copybook-name*' HAS NOT BEEN STANDARDIZED
Explanation: Copybooks are not standardized in the same run as a program.

Action: Process copybooks separately.

CQA12A THE 'BLOCK CONTAINS' CLAUSE FOR FILE NAME '*filename*' SHOULD INDICATE 0 IN AN OS ENVIRONMENT
Explanation: The BLOCK clause contains a value other than 0: BLOCK CONTAINS 0 provides for maximum run-time flexibility.

Action: Change the BLOCK clause to BLOCK CONTAINS O if an integer is not required.

CQA13A THE 'RECORD CONTAINS' CLAUSE FOR FILE NAME '*filename*' SHOULD NOT INDICATE 0 CHARACTERS

Explanation: An FD RECORD CONTAINS clause should indicate the actual length of the records associated with the file. This permits the compiler to cross-check the data record definitions with the FD.

Action: Specify the correct value for the clause.

CQA14W '*dataname-1* REDEFINES *dataname-2*' IS INCORRECT OR 'CASCADING'

Explanation: A REDEFINES object which does not meet the 1974 ANSI COBOL standard was encountered. The object of a REDEFINES clause must be the preceding data item at the current level that does not contain a REDEFINES clause. The macro set will attempt to correct the syntax.

Action: Verify the corrections produced by the macro. To eliminate cascading REDEFINES clauses, use the $CQA-REDEF YES option.

CQA15W THIS REDEFINES MAY BE INCORRECT DUE TO A COPYBOOK BEING UNAVAILABLE

Explanation: A REDEFINES refers to a data name that precedes an unprocessed COBOL COPY statement.

Action: The COPY statement can be unprocessed due to:

- an incorrect COPY statement. Check the statement.
- a missing copybook. Verify that the copybook exists.
- the non-specification of -INC. Specify the -INC translate-time option.
- the specification of COPY=IGNORE. Specify the COPY=ACTIVE or COPY=PASSIVE translate-time option.

CQA16A UNABLE TO DETERMINE IF THIS DATA ITEM '*dataname*' IS A GROUP OR ELEMENTARY ITEM

Explanation: An unavailable COPY element or a string-type macro was encountered, and the presence or absence of subordinate data items cannot be verified.

Action: Check the COPY statement or string-type macro, and correct necessary specifications.

CQA17A THESE DATA NAMES CANNOT BE REFERENCED. '*dataname-1* OF *dataname-2*'

Explanation: The displayed data names are not unique, even when qualified. It is possible that a FILLER item with subordinate data items has been encountered. The group item FILLER cannot be referenced.

Action: Check the FILLER item, and make necessary corrections.

CQA18A THIS PROGRAM CONTAINS 'OCCURS . . . DEPENDING ON . . .'
Explanation: An OCCURS...DEPENDING ON clause was encountered. Its use may cause difficulty in maintenance and debugging, and is therefore not recommended.

Action: Follow your site standards.

CQA19A THESE DATA NAMES MUST BE QUALIFIED. '*dataname-1* OF *dataname-2*'
Explanation: The displayed data can only be referenced when qualified.

Action: Change the names, and try again.

CQA20A THIS DATA ITEM '*dataname-1*' IS '*computational-1*', BUT IS NOT SYNCHRONIZED
Explanation: A dialect has been specified for a COBOL compiler earlier than the 1974 ANSI standard, where dataname is a usage data item which does not have a SYNCHRONIZED clause. This condition can inhibit compatibility between programs written for System/360 and System/370.

Action: Follow your site standards.

CQA21A THIS 'RENAMES' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE

Explanation: An occurrence of the RENAMES clause was encountered. Its use may cause problems during program maintenance.

Action: Follow your site standards. You can replace the RENAMES clause with a proper group name within the record structure.

CQA22E THERE IS NO SELECT STATEMENT FOR FILE NAME '*filename*'
Explanation: A required SELECT statement was omitted from an FD or SD file description.

Action: Enter the proper SELECT statement.

CQA23A THESE INDEX NAMES ARE NOT UNIQUE. '*index-name-1* OF *index-name-2*'
Explanation: The displayed index names are not unique; therefore, any reference to these names will generate a compiler error.

Action: Change the names before you rerun the program.

CQA24W THIS WORD '*word*' WAS DISCARDED
Explanation: An unrecognized word was diagnosed and deleted from the text.

Action: Specify a valid word, if needed.

CQA25W HAD TO ASSUME THAT THIS '*number*' IS A LEVEL NUMBER
Explanation: An out-of-place literal was diagnosed and assumed to be a level number.

Action: Correct the literal: the period that ends the previous statement may be missing.

CQA26W THIS IS AN ILLEGAL DATA NAME '*dataname*'
Explanation: The displayed *dataname* is a reserved word.

Action: Specify a valid name.

CQA27E THIS ELEMENTARY ITEM'S 'USAGE' '*dataname*' CONFLICTS WITH THE GROUP'S 'USAGE'
Explanation: A conflicting USAGE was found at the elementary group level when propagating a USAGE defined at the group level; the group USAGE was propagated.

Action: Follow your site standards.

CQA28A THE 'RECORD CONTAINS' CLAUSE IS A COMMENT
Explanation: The RECORD CONTAINS clause is unnecessary.

Action: Follow your site standards.

CQA29A THE ABOVE 'REDEFINES' HAS BEEN CORRECTED '*dataname-1* (NOW) REDEFINES *dataname-2*'
Explanation: The preceding REDEFINES error has been corrected.

Action: Check the changes.

CQA30A THIS ITEM '*dataname*' IS A GROUP ITEM IN A CONDITION
Explanation: The displayed *dataname* is a group item encountered in an IF statement. Incorrect results can occur if the subordinate items do not agree with the other comparand in CLASS, USAGE, and so on.

Action: Check your program results.

CQA31A THE 'ADD' STATEMENT ABOVE, CONTAINS ITEMS WITH CONFLICTING 'USAGE.' '*dataname-1*' (*subscript*)' USAGE IS DISPLAY '*dataname-2*' USAGE IS COMP-1
Explanation: The displayed data items have different USAGEs and cannot be manipulated.

Action: Convert the data item(s) to a common type.

CQA32A THE 'ADD' STATEMENT ABOVE, USES ITEMS REQUIRING POINT ALIGNMENT. '*dataname-1* (*subscript*) POINT *n1*' '*dataname-2* POINT *n2*'
Explanation: The displayed data items have different decimal point alignments that require padding and cannot be manipulated.

Action: Correct the decimal point alignment(s).

CQA33A THE 'ADD' STATEMENT ABOVE, USES ITEMS WITH DIFFERENT LENGTHS. '*dataname-1* (*subscript*) LENGTH *n1*' '*dataname-2* LENGTH *n2*
Explanation: The displayed data items have different lengths that require padding.

Action: Correct the length(s).

CQA34A THE ADD STATEMENT ABOVE, USES A 'USAGE DISPLAY' ITEM. '*dataname*'
Explanation: The displayed *dataname* is a DISPLAY item and was encountered in an arithmetic or conditional statement.

Action: Respecify the *dataname*.

CQA35A THE 'CORRESPONDING' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE.
Explanation: The use of CORRESPONDING in an ADD, SUBTRACT, or MOVE statement may cause problems in maintenance.

Action: Follow your site standards.

CQA36A 'ON SIZE ERROR' IS EXPENSIVE
Explanation: The use of ON SIZE ERROR is costly in the execution of programs and not recommended in a production environment.

Action: Follow your site standards.

CQA37A '*verb*' IS AN EXPENSIVE STATEMENT
Explanation: Using the displayed *verb* is costly in the execution of programs.

Action: Follow your site standards.

CQA38A THIS ITEM IS AN INEFFICIENT SUBSCRIPT. '*dataname* USAGE DISPLAY'
Explanation: Dataname is a subscript which is either a COMPUTATIONAL-3 or DISPLAY data item requiring type conversion for use.

Action: Follow your site standards.

CQA39A 'GO TO' MAY BE HARMFUL
Explanation: Excessive use of the GO TO statement is not recommended in a structured COBOL environment. Doing so may cause difficulty in program maintenance.

Action: Follow your site standards.

CQA40A FORWARD (DOWN) 'GO TO' LINE NUMBER=*number*, SEQUENCE NUMBER=*number*
Explanation: A GO TO statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure that follows the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

CQA41A BACKWARD (UP!) 'GO TO' LINE NUMBER=*number*, SEQUENCE NUMBER=*number*
Explanation: A GO TO statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure which precedes the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

CQA42A 'GO TO' GENERATED IN ALTER CONVERSION
Explanation: A GO TO statement was generated during the conversion of an ALTER statement.

Action: Follow your site standards.

CQA43A ALTER STATEMENT CONVERTED TO INITIALIZE A CONTROL VARIABLE
Explanation: An ALTER statement has been converted to a switch, a test, and a GO TO statement.

Action: Follow your site standards.

CQA44A THIS '*verb*' STATEMENT, DOES NOT HAVE A 'THRU' CLAUSE, AND IT SHOULD
Explanation: A THRU/THROUGH phrase is required for the SORT INPUT or OUTPUT PROCEDURE, MERGE OUTPUT PROCEDURE, or PERFORM statement displayed.

Action: Follow your site standards.

CQA45A THIS '*statement*' HAS A 'THRU' CLAUSE, AND IT SHOULD NOT
Explanation: The THRU/THROUGH phrase is not needed for the SORT INPUT or OUTPUT PROCEDURE, MERGE OUTPUT PROCEDURE, or PERFORM statement displayed.

Action: Follow your site standards.

CQA46A FORWARD (DOWN) PROCEDURE INVOCATION LINE NUMBER=*number*, SEQUENCE NUMBER=*number*
Explanation: A SORT, MERGE, PERFORM, or EXEC/EXECUTE CICS statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure that follows the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

CQA47A BACKWARD (UP!) PROCEDURE INVOCATION, LINE NUMBER=*number*, SEQUENCE NUMBER=*number*
Explanation: A SORT, MERGE, PERFORM, or EXEC/EXECUTE CICS statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure that precedes the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

CQA48A THIS '*verb*' STATEMENT USES THE 'FROM' CLAUSE
Explanation: An I/O statement with a FROM or INTO option was encountered.

Action: Follow your site standards.

CQA49A THIS '*verb*' STATEMENT DOES NOT USE THE 'FROM/INTO' CLAUSE
Explanation: An I/O statement without a FROM or INTO option was encountered.

Action: Follow your site standards.

CQA51A OPERATOR INTERVENTION IN '*statement*' IS A POOR OPERATING PROCEDURE
Explanation: An ACCEPT or DISPLAY statement requesting the use of the system console was encountered.

Action: Follow your site standards.

CQA52A THIS PROCEDURE HAS A PRIORITY NUMBER '*procedure-name-1* SECTION *number*'
Explanation: Program segmentation is more costly when running under a VS operating system than the equivalent paging overhead.

Action: Follow your site standards.

CQA53A DEBUGGING VERBS (*debugging-verb*) SHOULD NOT BE USED IN PRODUCTION PROGRAMS
Explanation: An occurrence of one or more of the following verbs was encountered:

READY TRACE
RESET TRACE
EXHIBIT
ON *literal*

Debugging statements should not exist in a production environment.

Action: Follow your site standards.

CQA54E THIS SUBSCRIPT SYNTAX IS INVALID. '*dataname* (*subscript*)'
Explanation: A subscript which is not valid COBOL syntax has been encountered.

Action: Correct the subscript syntax.

CQA55A THIS PROCEDURE MAY BE ACCOMPLISHING A SEQUENTIAL TABLE SEARCH '*procedure-name*'
Explanation: A sequential search of a table may have been encountered.

Action: A SEARCH statement should be specified for sequential table searches.

CQA57E THIS PARAGRAPH NAME IS A DUPLICATE '*procedure-name*'

Explanation: The displayed *procedure-name* is not unique. Numbering will not make it a unique name because CA-MetaCOBOL+ cannot determine which procedure is relevant.

Action: Change the name of at least one of the procedures in the section or program, then try again.

CQA58E THIS PROCEDURE NAME IS UNDEFINED '*procedure-name*'

Explanation: The displayed *procedure-name* cannot be located.

Action: Correct the error, then try again.

CQA60A THIS NAME IS TOO SHORT '*name*'

Explanation: The displayed '*name*' is shorter in length than the number specified by the $CQA-SHORT control statement.

Action: Follow your site standards.

CQA61A THIS PROCEDURE NAME, '*procedure-name-1*' HAS BEEN TRUNCATED '*proc-name-2*' DUE TO PROCEDURE NUMBERING

Explanation: The displayed *procedure-name-1* is longer than 30 characters when prefixed with a 4-digit sequence number and a hyphen. It has been truncated to *proc-name-2*, which is 30 characters in length.

Action: Follow your site standards.

CQA62A THIS DATA NAME IS UNDEFINED '*dataname*'

Explanation: The displayed *dataname* or Data Division section-name is not defined, and is therefore unavailable for Data Division mapping.

Action: Define *dataname* or remove its reference from the $CQA-MAP control statement.

CQA63A THIS PROCEDURE NAME IS UNDEFINED, '*procedure-name*' BUT MAY BE IN A COPYBOOK

Explanation: Reference was made to an undefined procedure. A warning (CQA63W) is issued when unprocessed COPY statements are also present. An error (CQA63E) is issued when COPY statements are not present, or COPY=ACTIVE is specified and all COPYs have been processed.

Action: Correct the warning by specifying COPY=ACTIVE. Correct the error by defining the procedure.

CQA64W AT LEAST ONE COPYBOOK IS UNAVAILABLE, RESULTS MAY BE INCORRECT
Explanation: One or more COPY elements were unavailable for processing.

Action: Specify the COPY=ACTIVE or -INC translate-time options.

CQA65A THE FOLLOWING IS AN IBM EXTENSION '*extension*'
Explanation: The displayed *extension* is an IBM extension to the 1974 ANSI COBOL standard. It has been retained to ensure upward portability from the Version 4 compiler, available under non-VS operating systems.

Action: Follow your site standards.

CQA66W A 'NULL' GO TO STATEMENT HAS BEEN DISCARDED
Explanation: A GO TO statement without a destination was discarded. CA-MetaCOBOL+ assumes an ALTER statement replaced the GO TO with an IF statement and a GO TO statement.

Action: Check the conversion to make sure program control is not adversely affected, then rerun.

CQA67A THE DATA DIVISION HAS BEEN COMPLETED TO SUPPORT DATA MAPPING
Explanation: Regardless of the absence of a Procedure Division header, CQA completed the internal tables, thus permitting mapping.

Action: No action required. However, a Procedure Division header can be entered.

CQA68W THIS PROGRAM HAS TOO MANY PROCEDURE NAMES, GIVEN THE '$CQA-SEQN' PARAMETER SPECIFIED (*number*). PROCEDURE NAME NUMBERING HAS BEEN DISCONTINUED
Explanation: The number of paragraphs and the paragraph numbering increment have caused the renumbering process to overflow.

Action: Specify a smaller value for the $CQA-SEQN parameter.

CQA69E THIS PROCEDURE NAME IS ALSO A DATA NAME '*procedure-name*'
Explanation: The displayed *procedure-name* has already been used as a data name; no renumbering will be done.

Action: Change the paragraph or data name to make them unique.

CQA70E MORE THAN 5000 PARAGRAPHS ARE REFERENCED PRIOR TO THEIR DEFINITIONS. PARAGRAPH NUMBERING HAS BEEN DISCONTINUED
Explanation: All of CA-MetaCOBOL+'s OUT-OF-LINE markers (&MARKER) have been utilized. A marker is required for each of the following:

- reference to a paragraph or section which precedes the definition of the paragraph or section
- GO TO paragraph or section
- ALTER statement

Action: Follow your site standards.

CQA71A 'REDEFINES FILLER' WAS NOT CHANGED

Explanation: REDEFINES FILLER is vague.

Action: Provide a name for the entity you want to redefine, then try again.

CQA72A THIS IS AN INVALID LEVEL NUMBER, '*nn*'

Explanation: The displayed level number is invalid. Valid values are 1-49, 66, 77, or 88.

Action: Check your program, make necessary corrections, and rerun.

CQA73E A STRING MACRO CALL ->*string macroname*<- HAS BEEN IGNORED

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL+ Support.

CQA74A A MISSING PERIOD HAS BEEN RESTORED

Explanation: Required punctuation was inserted by CA-MetaCOBOL+.

Action: Check your program, then try again.

CQA75E AN 'ALTER' TARGET IS NOT A 'GO TO' PARAGRAPH. LINE NUMBER = *number* SEQUENCE NUMBER = *number*

Explanation: The target of an ALTER statement must be a GO TO statement.

Action: Check your program, make sure the target of the ALTER statement is a GO TO statement, then try again.

CQA77W '$CQA-*name value-1*' IS INVALID. DEFAULT STATUS (*value-2*) UNCHANGED

Explanation: An invalid value was specified for the $CQA-*name* statement. Valid values are Y (YES) or N (NO). The statement has accepted its default.

Action: Check your program logic. If the default is appropriate, try to rerun the program. If the default is not appropriate, change the value before you try to rerun the program.

CQA78E COBOL QUALITY ASSURANCE REQUIRES 'OPTION DIALECT=VO, VD, WO, OR WD'

Explanation: An invalid value was specified for the CA-MetaCOBOL DIALECT option. Valid values are 'V' or 'W'.

Action: Specify a valid value, and try again.

CQA79W '$CQA-RELATIONS *value-1*' IS INVALID. DEFAULT STATUS (*value-2*) UNCHANGED.

Explanation: An invalid value was specified for a $CQA-RELATIONS *name* statement. Valid values are S (SHORT) or L (LONG). The statement has accepted its default.

Action: Check your program logic. If the default is appropriate, try to rerun the program. If the default is not appropriate, change the value before you try to rerun the program.

CQA80E DATADICTIONARY INTERFACE FAILURE. SIGNON ERROR

Explanation: The CA-DATADICTIONARY interface has failed as specified.

Action: Check the JCL, DATAVIEW statement, program-name, DDID, PSTAT and PVER options.

CQA81W '$CQA-SHORT *value*' IS INVALID. DEFAULT STATUS (*nn*) UNCHANGED

Explanation: An invalid value was specified for a $CQA-SHORT control statement.

Action: The value must be an integral numeric literal less than 30 or 0; correct the $CQA-SHORT control statement.

CQA82W '$CQA-SEQN *value*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED

Explanation: An invalid value was specified for a $CQA-SEQN control statement.

Action: The value must be an integral numeric literal less than 1000; correct the $CQA-SEQN control statement.

CQA83W 'CQA-MAP' SPECIFIES TOO MANY NAMES. ONLY THE FIRST 10 WILL BE USED

Explanation: More than 10 data structures were entered in the $CQA-MAP control statement(s). Only the first 10 will be mapped.

Action: Correct the $CQA-MAP control statement(s).

CQA90W '$CQA-INCR *value*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED.

Explanation: An invalid value was encountered in a $CQA-INCR control statement.

Action: The value must be a numeric literal or 0; correct the $CQA-INCR control statement.

CQA92W THIS PROCEDURE DIVISION CONTAINS COPY STATEMENTS.
PROCEDURE NAMES IN THE COPYBOOKS HAVE NOT BEEN NUMBERED

Explanation: The COPY=PASSIVE or COPY=IGNORE translate-time option has been specified, preventing the processing of copybook contents. The second clause is issued only when procedure renumbering is being performed.

Action: Specify COPY=ACTIVE.

CQA93E THIS IS A DUPLICATE SECTION NAME. '*section-name*'

Explanation: A section or paragraph name is used more than once.

Action: Section names must be unique; paragraph names must be unique or qualified.

CQA94A THIS IS A NON-UNIQUE PARAGRAPH NAME '*paragraph-name*'

Explanation: The PROCEDURE DIVISION contains a non-unique paragraph name.

Action: Specify a number greater than 0 in the $CQA-SEQN control statement or rename the paragraph name.

CQA95E THIS CICS STATEMENT IS INVALID

Explanation: An incorrectly terminated EXEC/EXECUTE CICS statement was located in the Procedure Division.

Action: Review syntax and correct the statement.

CQA98E $PDX CODE GENERATION ERROR.
STATE=0
SYMBOL=NA

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL+ Support.

CQA99W '$CQA-*clause*-COLUMN-*nn*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED.

Explanation: An improper integer was specified with the $CQA-*clause*-COLUMN control statement; columnar alignment is ignored.

Action: Correct the $CQA-*clause*-COLUMN control statement.

B.1.2 CQA/DCQA Internal Error Diagnostic

The following diagnostic is not numbered. It is issued only when CQA discovers what appears to be an internal logic error. The diagnostic and all supporting documentation should be supplied to your Computer Associates representative for corrective action.

CQA INTERNAL ERROR - FD

Explanation: CQA has discovered its own error in the FD macro.

Action: Contact your Computer Associates representative.

B.2 SQA/DSQA NOTE Diagnostics

NOTE diagnostics contained within SQA begin with the following prefix:

Format:

SQA*nnz*

SQA represents the SQA diagnostic abbreviation.

nn represents the diagnostic number and is unique within SQA.

z represents one of the following severity codes:

E an error requiring correction and retranslation; associated with a condition code of 12.

W a warning, possibly requiring correction and retranslation; associated with a condition code of 8.

A an advisory comment; associated with a condition code of 4.

This section provides an explanation for each NOTE diagnostic produced by SQA. Corrective action is provided for error and warning diagnostics; corrective action for advisory diagnostics is user defined unless otherwise indicated.

B.2.1 Standard SQA/DSQA Messages

SQA01A THIS 'SAME AREA' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE

Explanation: The SAME AREA clause places several files within the same buffer in order to save storage. As a result, an overlay of records may occur which will produce data not expected by the program.

Action: Check program results.

SQA02E TOO MANY NESTED SEARCH STATEMENTS

Explanation: More than 100 SEARCH statements have been nested.

Action: Respecify values for the variables that support this check, or follow your site standards.

SQA04A THIS LONG RELATION '*long statement*' HAS BEEN CHANGED TO A SHORT RELATION AS FOLLOWS: '*short statement*'

Explanation: The $SQA-RELATIONS SHORT option standardizes all relational operators to a short form, for example, IS EQUAL TO becomes =. IF, PERFORM, SEARCH, and START conditions are standardized. For DSQA, the DL WHERE statement is also standardized.

Action: No action is required.

SQA05A THIS SHORT RELATION '*short statement*' HAS BEEN CHANGED TO A LONG RELATION AS FOLLOWS: '*long statement*'

Explanation: The $SQA-RELATIONS LONG option standardizes all relational operators to a long form, for example, = becomes IS EQUAL TO. IF, PERFORM, SEARCH, and START conditions are standardized. For DSQA, the DL WHERE statement is also standardized.

Action: No action is required.

SQA06A THIS ELEMENTARY ITEM HAS A 'VALUE' '*dataname*' WHICH CONFLICTS WITH A GROUP ITEM THAT ALSO HAS A 'VALUE' CLAUSE

Explanation: An elementary item with a VALUE clause is subordinate to a group level item that also has a VALUE clause.

Action: Remove this conflict.

SQA07A THIS 'BLOCK CONTAINS' CLAUSE IS NOT NEEDED. '*filename*' IS A VSAM FILE

Explanation: The BLOCK CONTAINS clause is ignored when the file is a VSAM file.

Action: No action is required. However, you can remove the BLOCK CONTAINS clause.

SQA09E INVALID $TACK FUNCTION CODE

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL Support.

SQA10E THE GENERATED LEVEL NUMBER '*number*' FOR '*dataname*' IS INCORRECT. THE $SQA-INCR PARAMETER IS TOO LARGE

Explanation: A level number greater than 49 was generated during Data Division level number normalization.

Action: Specify a smaller increment on the $SQA-INCR control statement.

SQA11W THE COPYBOOK '*copybook-name*' HAS NOT BEEN STANDARDIZED

Explanation: Copybooks are not standardized in the same run as a program.

Action: Process copybooks separately.

SQA12A THE 'BLOCK CONTAINS' CLAUSE FOR FILE NAME '*filename*' SHOULD INDICATE 0 IN AN OS ENVIRONMENT

Explanation: The BLOCK clause contains a value other than 0: BLOCK CONTAINS 0 provides for maximum run-time flexibility.

Action: Change the BLOCK clause to BLOCK CONTAINS 0 if an integer is not required.

SQA13A THE 'RECORD CONTAINS' CLAUSE FOR FILE NAME '*filename*' SHOULD NOT INDICATE 0 CHARACTERS

Explanation: An FD RECORD CONTAINS clause should indicate the actual length of the records associated with the file. This permits the compiler to cross-check the data record definitions with the FD.

Action: Specify the correct value for the clause.

SQA14W '*dataname-1* REDEFINES *dataname-2*' IS INCORRECT OR 'CASCADING'

Explanation: A REDEFINES object which does not meet the 1974 ANSI COBOL standard was encountered. The object of a REDEFINES clause must be the preceding data item at the current level that does not contain a REDEFINES clause. The macro set will attempt to correct the sytnax.

Action: Verify the corrections produced by the macro. To eliminate cascading REDEFINES clauses, use the $SQA-REDEF YES option.

SQA15W THIS REDEFINES MAY BE INCORRECT DUE TO A COPYBOOK BEING UNAVAILABLE

Explanation: A REDEFINES refers to a data name that precedes an unprocessed COBOL COPY statement.

Action: The COPY statement may be unprocessed due to:

- an incorrect COPY statement. Check the statement.
- a missing copybook. Verify that the copybook exists.
- the non-specification of -INC. Specify the -INC translate-time option.
- the specification of COPY=IGNORE. Specify the COPY=ACTIVE or COPY=PASSIVE translate-time option.

SQA16A UNABLE TO DETERMINE IF THIS DATA ITEM '*dataname*' IS A GROUP OR ELEMENTARY ITEM

Explanation: An unavailable COPY element or a string-type macro was encountered, and the presence or absence of subordinate data items cannot be verified.

Action: Check the COPY statement or string-type macro, and correct necessary specifications.

SQA17A THESE DATA NAMES CANNOT BE REFERENCED. '*dataname-1* OF *dataname-2*'

Explanation: The displayed data names are not unique, even when qualified. It is possible that a FILLER item with subordinate data items has been encountered: The group item FILLER cannot be referenced.

Action: Check the FILLER item, and make necessary corrections.

SQA18A THIS PROGRAM CONTAINS 'OCCURS . . . DEPENDING ON . . .'

Explanation: An OCCURS...DEPENDING ON clause was encountered. Its use may cause difficulty in maintenance and debugging, and is therefore not recommended.

Action: Follow your site standards.

SQA19A THESE DATA NAMES MUST BE QUALIFIED. '*dataname-1* OF *dataname-2*'

Explanation: The displayed datanames can only be referenced when qualified.

Action: Change the names, and try again.

SQA20A THIS DATA ITEM '*dataname-1*' IS '*computational-1*', BUT IS NOT SYNCHRONIZED

Explanation: A dialect was specified for a COBOL compiler earlier than the 1974 ANSI standard, where dataname is a usage data item which does not have a SYNCHRONIZED clause. This condition can inhibit compatibility between programs written for System/360 and System/370.

Action: Follow your site standards.

SQA21A THIS 'RENAMES' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE

Explanation: An occurrence of the RENAMES clause was encountered. Its use may cause problems during program maintenance.

Action: Follow your site standards. You can replace the RENAMES clause with a proper group name within the record structure.

SQA22E THERE IS NO SELECT STATEMENT FOR FILE NAME '*filename*'

Explanation: A required SELECT statement was omitted from an FD or SD file description.

Action: Enter the proper SELECT statement.

SQA23A THESE INDEX NAMES ARE NOT UNIQUE. '*index-name-1* OF *index-name-2*'

Explanation: The displayed index names are not unique; therefore, any reference to these names will generate a compiler error.

Action: Change the names before you rerun the program.

SQA24W THIS WORD '*word*' WAS DISCARDED

Explanation: An unrecognized word was diagnosed and deleted from the text.

Action: If needed, specify a valid word.

SQA25W HAD TO ASSUME THAT THIS '*number*' IS A LEVEL NUMBER

Explanation: An out-of-place literal was diagnosed and assumed to be a level number.

Action: Correct the literal: the period that ends the previous statement may be missing.

SQA26W THIS IS AN ILLEGAL DATA NAME '*dataname*'

Explanation: The displayed *dataname* is a reserved word.

Action: Specify a valid name.

SQA27E THIS ELEMENTARY ITEM'S 'USAGE' '*dataname*' CONFLICTS WITH THE GROUP'S 'USAGE'

Explanation: A conflicting USAGE was found at the elementary group level when propagating a USAGE defined at the group level; the group USAGE was propagated.

Action: Follow your site standards.

SQA28A THE 'RECORD CONTAINS' CLAUSE IS A COMMENT

Explanation: The RECORD CONTAINS clause is unnecessary.

Action: Follow your site standards.

SQA29A THE ABOVE 'REDEFINES' HAS BEEN CORRECTED '*dataname-1* (NOW) REDEFINES *dataname-2*'

Explanation: The preceding REDEFINES error was corrected.

Action: Check the changes.

SQA30A THIS ITEM '*dataname*' IS A GROUP ITEM IN A CONDITION

Explanation: The displayed *dataname* is a group item encountered in an IF statement. Incorrect results can occur if the subordinate items do not agree with the other comparand in CLASS, USAGE, and so on.

Action: Check your program results.

SQA31A THE 'ADD' STATEMENT ABOVE, CONTAINS ITEMS WITH CONFLICTING 'USAGE.' '*dataname-1*' (*subscript*)' USAGE IS DISPLAY '*dataname-2*' USAGE IS COMP-1

Explanation: The displayed data items have different USAGEs and cannot be manipulated.

Action: Convert the data item(s) to a common type.

SQA32A THE 'ADD' STATEMENT ABOVE, USES ITEMS REQUIRING POINT ALIGNMENT. '*dataname-1* POINT*n1*' '*dataname-2* POINT*n2*'

Explanation: The displayed data items have different decimal point alignments that require padding and cannot be manipulated.

Action: Correct the decimal point alignment(s).

SQA33A THE 'ADD' STATEMENT ABOVE, USES ITEMS WITH DIFFERENT LENGTHS. '*dataname-1* LENGTH*n1*' '*dataname-2* LENGTH*n2*

Explanation: The displayed data items have different lengths that require padding.

Action: Correct the length(s).

SQA34A THE ADD STATEMENT ABOVE, USES A 'USAGE DISPLAY' ITEM. '*dataname*'

Explanation: The displayed *dataname* is a DISPLAY item and was encountered in an arithmetic or conditional statement.

Action: Respecify the *dataname*.

SQA35A THE 'CORRESPONDING' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE

Explanation: The use of CORRESPONDING in an ADD, SUBTRACT, or MOVE statement can cause problems in maintenance.

Action: Follow your site standards.

SQA36A 'IF SIZE ERROR' IS EXPENSIVE

Explanation: The use of ON SIZE ERROR is costly in the execution of programs and not recommended in a production environment.

Action: Follow your site standards.

SQA37A '*verb*' IS AN EXPENSIVE STATEMENT

Explanation: Using the displayed *verb* is costly in the execution of programs.

Action: Follow your site standards.

SQA38A THIS ITEM IS AN INEFFICIENT SUBSCRIPT. '*dataname* USAGE DISPLAY'

Explanation: Dataname is a subscript which is either a COMPUTATIONAL-3 or DISPLAY data item requiring type conversion for use.

Action: Follow your site standards.

SQA46A FORWARD (DOWN) PROCEDURE INVOCATION LINE NUMBER=*number*, SEQUENCE NUMBER=*number*

Explanation: A SORT, MERGE, PERFORM, or EXEC/EXECUTE CICS statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure that follows the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

SQA47A BACKWARD (UP!) PROCEDURE INVOCATION, LINE NUMBER=number, SEQUENCE NUMBER=number

Explanation: A SORT, MERGE, PERFORM, or EXEC/EXECUTE CICS statement, coded at CA-MetaCOBOL+ LINE NUMBER=number, COBOL SEQUENCE NUMBER=number, refers to a procedure that precedes the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

SQA48A THIS '*verb*' STATEMENT USES THE 'FROM' CLAUSE

Explanation: An I/O statement with a FROM or INTO option was encountered.

Action: Follow your site standards.

SQA49A THIS '*verb*' STATEMENT DOES NOT USE THE 'FROM/INTO' CLAUSE

Explanation: An I/O statement without a FROM or INTO option was encountered.

Action: Follow your site standards.

SQA50E LOCAL DATA DEFINED IN 'START DATA' IS NOT TERMINATED WITH AN 'END DATA'

Explanation: A 'START DATA' statement is missing its required 'END DATA' statement.

Action: Enter the required 'END DATA' statement, and try again.

SQA51A OPERATOR INTERVENTION IN '*statement*' IS A POOR OPERATING PROCEDURE

Explanation: An ACCEPT or DISPLAY statement requesting the use of the system console was encountered.

Action: Follow your site standards.

SQA52A THIS PROCEDURE HAS A PRIORITY NUMBER '*procedure-name-1* SECTION *nn*'

Explanation: Program segmentation is more costly when running under a VS operating system than the equivalent paging overhead.

Action: Follow your site standards.

SQA53A DEBUGGING VERBS (*debugging-verb*) SHOULD NOT BE USED IN PRODUCTION PROGRAMS

Explanation: An occurrence of one or more of the following verbs was encountered:

READY TRACE
RESET TRACE
EXHIBIT
ON *literal*

Debugging statements should not exist in a production environment.

Action: Follow your site standards.

SQA54E THIS SUBSCRIPT SYNTAX IS INVALID. '*dataname* (*subscript*)'

Explanation: A subscript which is not valid COBOL syntax has been encountered.

Action: Correct the subscript syntax.

SQA57E THIS PARAGRAPH NAME IS A DUPLICATE '*procedure-name*'

Explanation: The displayed *procedure-name* is not unique. Numbering will not make it a unique name because CA-MetaCOBOL+ cannot determine which procedure is relevant.

Action: Change the name of at least one of the procedures in the section or program, then try again.

SQA58E THIS PROCEDURE NAME IS UNDEFINED '*procedure-name*'

Explanation: The displayed *procedure-name* cannot be located.

Action: Correct the error, then try again.

SQA60A THIS NAME IS TOO SHORT '*name*'

Explanation: The displayed '*name*' is shorter in length than the number specified by the $CQA-SHORT control statement.

Action: Follow your site standards.

SQA61A THIS PROCEDURE NAME, '*procedure-name-1*' HAS BEEN TRUNCATED '*proc-name-2*' DUE TO PROCEDURE NUMBERING

Explanation: The displayed *procedure-name-1* is longer than 30 characters when prefixed with a 4-digit sequence number and a hyphen. It has been truncated to *proc-name-2*, which is 30 characters in length.

Action: Follow your site standards.

SQA62A THIS DATA OR FLAG NAME IS UNDEFINED '*dataname* (*nn*)' (DEFINITION MAY BE IN NEXT MODULE)

Explanation: The displayed *dataname* or Data Division section-name is not defined, and is therefore unavailable for Data Division mapping.

Action: Define *dataname* or remove its reference from the $SQA-MAP control statement.

SQA63A THIS PROCEDURE NAME IS UNDEFINED, '*procedure-name*'

Explanation: Reference was made to an undefined procedure. A warning (SQA63W) is issued when unprocessed COPY statements are also present. An error (SQA63E) is issued when COPY statements are not present or when COPY=ACTIVE is specified and all COPYs have been processed.

Action: To correct the warning, specify COPY=ACTIVE; to correct the error, define the procedure.

SQA64W AT LEAST ONE COPYBOOK IS UNAVAILABLE, RESULTS MAY BE INCORRECT

Explanation: One or more COPY elements were unavailable for processing.

Action: Specify the COPY=ACTIVE or -INC translate-time options.

SQA65A THE FOLLOWING IS AN IBM EXTENSION '*extension*'

Explanation: The displayed *extension* is an IBM extension to the 1974 ANSI COBOL standard. It has been retained to ensure upward portability from the Version 4 compiler, available under non-VS operating systems.

Action: Follow your site standards.

SQA67A THE DATA DIVISION HAS BEEN COMPLETED TO SUPPORT DATA MAPPING

Explanation: Regardless of the absence of a Procedure Division header, SQA completed the internal tables, thus permitting mapping.

Action: No action required. However, a Procedure Division header can be entered.

SQA68W THIS PROGRAM HAS TOO MANY PROCEDURE NAMES, GIVEN THE '$SQA-SEQN' PARAMETER SPECIFIED (*number*). PROCEDURE NAME NUMBERING HAS BEEN DISCONTINUED

Explanation: The number of paragraphs and the paragraph numbering increment have caused the renumbering process to overflow.

Action: Specify a smaller value for the $CQA-SEQN parameter.

SQA69E THIS PROCEDURE NAME IS ALSO A DATA NAME '*procedure-name*'

Explanation: The displayed *procedure-name* has already been used as a data name; no renumbering will be done.

Action: Change the paragraph or data name to make them unique.

SQA70E MORE THAN 5000 PARAGRAPHS ARE REFERENCED PRIOR TO THEIR DEFINITIONS. PARAGRAPH NUMBERING HAS BEEN DISCONTINUED

Explanation: All of CA-MetaCOBOL+'s OUT-OF-LINE markers (&MARKER) have been utilized. A marker is required for each:

- reference to a paragraph or section which precedes the definition of the paragraph or section
- GO TO paragraph or section
- ALTER statement

Action: Follow your site standards.

SQA71A 'REDEFINES FILLER' WAS NOT CHANGED

Explanation: REDEFINES FILLER is vague.

Action: Provide a name for the entity you want to redefine, then try again.

SQA72A THIS IS AN INVALID LEVEL NUMBER, '*nn*'

Explanation: The displayed level number is invalid. Valid values are 1-49, 66, 77, or 88.

Action: Check your program, make necessary corrections, and rerun.

SQA73E A STRING MACRO CALL ->*string macroname*<- HAS BEEN IGNORED

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL+ Support.

SQA74A A MISSING PERIOD HAS BEEN RESTORED

Explanation: Required punctuation was inserted by CA-MetaCOBOL+

Action: Check your program, then try again.

SQA77W '$SQA-*name value-1*' IS INVALID. DEFAULT STATUS (*value-2*) UNCHANGED

Explanation: An invalid value was specified for a $SQA-*name* statement. Valid values are Y (YES) or N (NO). The statement has accepted its default.

Action: Check your program logic. If the default is appropriate, try to rerun the program. If the default is not appropriate, change the value before you try to rerun the program.

SQA78E COBOL/SP QUALITY ASSURANCE REQUIRES 'OPTION DIALECT=VO, VD, WO, OR WD'

Explanation: An invalid value was specified for the CA-MetaCOBOL DIALECT option. Valid values are 'V' or 'W'.

Action: Specify a valid value, and try again.

SQA79W '$SQA-RELATIONS *value-1*' IS INVALID. DEFAULT STATUS (*value-2*) UNCHANGED.

Explanation: An invalid value was specified for a $SQA-RELATIONS *name* statement. Valid values are S (SHORT) or L (LONG). The statement has accepted its default.

Action: Check your program logic. If the default is appropriate, try to rerun the program; if the default is not appropriate, change the value before you try to rerun the program.

SQA80E DATADICTIONARY INTERFACE FAILURE. SIGNON ERROR

Explanation: The Datadictionary interface has failed as specified.

Action: Check the JCL, DATAVIEW statement, program-name, DDID, PSTAT and PVER options.

SQA81W '$SQA-SHORT *value*' IS INVALID. DEFAULT STATUS (*nn*) UNCHANGED

Explanation: An invalid value was encountered in a $SQA-SHORT control statement.

Action: The value must be an integral numeric literal less than 30 or 0; correct the $SQA-SHORT control statement.

SQA82W '$SQA-SEQN *value*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED

Explanation: An invalid value was encountered in a $SQA-SEQN control statement.

Action: The value must be an integral numeric literal less than 1000; correct the $SQA-SEQN control statement.

SQA83W 'SQA-MAP' SPECIFIES TOO MANY NAMES. ONLY THE FIRST 10 WILL BE USED

Explanation: More than 10 data structures were entered in the $SQA-MAP control statement(s). Only the first 10 will be mapped.

Action: Correct the $SQA-MAP control statement.

SQA84E STRUCTURE ERROR - '*structure-code-1*,' CURRENT STRUCTURE IS '*structure-code-2*' ON LINE *number*

Explanation: The specified '*structure-code-1*' does not agree with the current control structure.

Action: Correct the structure error, and try again.

SQA90W '$SQA-INCR *value*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED.

Explanation: An invalid value was encountered in a $SQA-INCR control statement.

Action: The value must be a numeric literal or 0; correct the $SQA-INCR control statement.

SQA92W THIS PROCEDURE DIVISION CONTAINS COPY STATEMENTS. PROCEDURE NAMES IN THE COPYBOOKS HAVE NOT BEEN NUMBERED

Explanation: The COPY=PASSIVE or COPY=IGNORE translate-time option has been specified, preventing the processing of copybook contents. The second clause is issued only when procedure renumbering is being performed.

Action: Specify COPY=ACTIVE.

SQA93E THIS IS A DUPLICATE SECTION NAME. '*section-name*'

Explanation: A section or paragraph name is used more than once.

Action: Section names must be unique; paragraph names must be unique or qualified.

SQA94A THIS IS A NON-UNIQUE PARAGRAPH NAME '*paragraph-name*'

Explanation: The PROCEDURE DIVISION contains a non-unique paragraph name.

Action: Specify a number greater than 0 in the $SQA-SEQN control statement or rename the paragraph name.

SQA95E THIS CICS STATEMENT IS INVALID

Explanation: An incorrectly terminated EXEC/EXECUTE CICS statement was located in the Procedure Division.

Action: Review syntax and correct the statement.

SQA98E $PDX CODE GENERATION ERROR.
STATE=0
SYMBOL=NA

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL+ Support.

SQA99W '$SQA-*clause*-COLUMN-*nn*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED.

Explanation: An improper integer was specified with the $SQA-*clause*-COLUMN control statement; columnar alignment is ignored.

Action: Correct the $SQA-*clause*-COLUMN control statement.

B.2.2 SQA/DSQA Internal Error Diagnostics

The following diagnostic is not numbered. It is issued only when SQA discovers what appears to be an internal logic error. The diagnostic and all supporting documentation should be supplied to your Computer Associates representative for corrective action.

SQA INTERNAL ERROR-FD

Explanation: SQA has discovered its own error in the FD macro.

Action: Contact your Computer Associates representative.

B.3 VQA/DVQA NOTE Diagnostics

NOTE diagnostics contained within VQA begin with the following prefix:

Format:

VQA*nnz*

VQA represents the VQA diagnostic abbreviation.

nn represents the diagnostic number unique within VQA.

z represents one of the following severity codes:

E an *error* requiring correction and retranslation; associated with a condition code of 12.

W a *warning*, possibly requiring correction and retranslation; associated with a condition code of 8.

A an *advisory* comment; associated with a condition code of 4.

This section provides an explanation for each NOTE diagnostic produced by VQA. Corrective action is provided for error and warning diagnostics. Corrective action for advisory diagnostics is user defined unless otherwise indicated.

B.3.1 Standard VQA/DVQA Messages

VQA01A THIS 'SAME AREA' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE

Explanation: The SAME AREA clause places several files within the same buffer in order to save storage. As a result, an overlay of records may occur which will produce data not expected by the program.

Action: Check program results.

VQA02E TOO MANY NESTED SEARCH AND EVALUATE STATEMENTS

Explanation: More than 100 SEARCH statements have been nested.

Action: Respecify values for the variables that support this check, or follow your site standards.

VQA03A THIS 'GO TO ... DEPENDING ON ...' STATEMENT MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE

Explanation: The GO TO . . . DEPENDING ON . . . statement may cause serious problems during program maintenance.

Action: Be careful with the control variable, and document program flow control for future reference.

VQA04A THIS LONG RELATION '*long statement*' HAS BEEN CHANGED TO A SHORT RELATION AS FOLLOWS: '*short statement*'

Explanation: The $VQA-RELATIONS SHORT option standardizes all relational operators to a short form, for example, IS EQUAL TO becomes =. IF, PERFORM, SEARCH, and START conditions are standardized. For DVQA, the DL WHERE statement is also standardized.

Action: No action is required.

VQA05A THIS SHORT RELATION '*short statement*' HAS BEEN CHANGED TO A LONG RELATION AS FOLLOWS: '*long statement*'

Explanation: The $VQA-RELATIONS LONG option standardizes all relational operators to a long form, for example, = becomes IS EQUAL TO. IF, PERFORM, SEARCH, and START conditions are standardized. For DVQA, the DL WHERE statement is also standardized.

Action: No action is required.

VQA06A THIS ELEMENTARY ITEM HAS A 'VALUE' '*dataname*' WHICH CONFLICTS WITH A GROUP ITEM THAT ALSO HAS A 'VALUE' CLAUSE

Explanation: An elementary item with a VALUE clause is subordinate to a group level item that also has a VALUE clause.

Action: Remove this conflict.

VQA07A THIS 'BLOCK CONTAINS' CLAUSE IS NOT NEEDED. '*filename*' IS A VSAM FILE

Explanation: The BLOCK CONTAINS clause is ignored when the file is a VSAM file.

Action: No action is required. However, you can remove the BLOCK CONTAINS clause.

VQA09E INVALID $TACK FUNCTION CODE

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL Support.

VQA10E THE GENERATED LEVEL NUMBER '*nn*' FOR '*dataname*' IS INCORRECT. THE $VQA-INCR PARAMETER IS TOO LARGE

Explanation: A level number greater than 49 was generated during Data Division level number normalization.

Action: Specify a smaller increment on the $VQA-INCR control statement.

VQA11W THE COPYBOOK '*copybook-name*' HAS NOT BEEN STANDARDIZED

Explanation: Copybooks are not standardized in the same run as a program.

Action: Process copybooks separately.

VQA12A THE 'BLOCK CONTAINS' CLAUSE FOR FILE NAME '*filename*' SHOULD INDICATE 0 IN AN OS ENVIRONMENT

Explanation: The BLOCK clause contains a value other than 0: BLOCK CONTAINS 0 provides for maximum run-time flexibility.

Action: Change the BLOCK clause to BLOCK CONTAINS 0 if an integer is not required.

VQA13A THE 'RECORD CONTAINS' CLAUSE FOR FILE NAME '*filename*' SHOULD NOT INDICATE 0 CHARACTERS

Explanation: An FD RECORD CONTAINS clause should indicate the actual length of the records associated with the file. This permits the compiler to cross-check the data record definitions with the FD.

Action: Specify the correct value for the clause.

VQA14W '*dataname-1* REDEFINES *dataname-2*' IS INCORRECT OR 'CASCADING'

Explanation: A REDEFINES object which does not meet the 1974 ANSI COBOL standard was encountered. The object of a REDEFINES clause must be the preceding data item at the current level that does not contain a REDEFINES clause. The macro set will attempt to correct the syntax.

Action: Verify the corrections produced by the macro. To eliminate cascading REDEFINES clauses, use the $VQA-REDEF YES option.

VQA15W THIS REDEFINES MAY BE INCORRECT DUE TO A COPYBOOK BEING UNAVAILABLE

Explanation: A REDEFINES refers to a data name that precedes an unprocessed COBOL COPY statement.

Action: The COPY statement may be unprocessed due to:

- an incorrect COPY statement. Check the statement.
- a missing copybook. Verify that the copybook exists.
- the non-specification of -INC. Specify the -INC translate-time option.
- the specification of COPY=IGNORE. Specify the COPY=ACTIVE or COPY=PASSIVE translate-time option.

VQA16A UNABLE TO DETERMINE IF THIS DATA ITEM '*dataname*' IS A GROUP OR ELEMENTARY ITEM

Explanation: An unavailable COPY element or a string-type macro was encountered, and the presence or absence of subordinate data items cannot be verified.

Action: Check the COPY statement or string-type macro, and correct necessary specifications.

VQA17A THESE DATA NAMES CANNOT BE REFERENCED. '*dataname-1* OF *dataname-2*'

Explanation: The displayed data names are not unique, even when qualified. It's possible that a FILLER item with subordinate data items has been encountered: The group item FILLER cannot be referenced.

Action: Check the FILLER item, and make necessary corrections.

VQA18A THIS PROGRAM CONTAINS 'OCCURS . . . DEPENDING ON . . .'

Explanation: An OCCURS...DEPENDING ON clause was encountered. Its use may cause difficulty in maintenance and debugging, and is therefore not recommended.

Action: Follow your site standards.

VQA19A THESE DATA NAMES MUST BE QUALIFIED. '*dataname-1* OF *dataname-2*'

Explanation: The displayed *dataname* can only be referenced when qualified.

Action: Change the names, and try again.

VQA20A THIS DATA ITEM '*dataname-1*' IS '*computational-1*', BUT IS NOT SYNCHRONIZED

Explanation: A dialect was specified for a COBOL compiler earlier than the 1974 ANSI standard, where dataname is a usage data item which does not have a SYNCHRONIZED clause. This condition can inhibit compatibility between programs written for System/360 and System/370.

Action: Follow your site standards.

VQA21A THIS 'RENAMES' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE

Explanation: An occurrence of the RENAMES clause was encountered. Its use may cause problems during program maintenance.

Action: Follow your site standards. You can replace the RENAMES clause with a proper group name within the record structure.

VQA22E THERE IS NO SELECT STATEMENT FOR FILE NAME '*filename*'

Explanation: A required SELECT statement was omitted from an FD or SD file description.

Action: Enter the proper SELECT statement.

VQA23A THESE INDEX NAMES ARE NOT UNIQUE. '*index-name-1* OF *index-name-2*'

Explanation: The displayed index names are not unique; therefore, any reference to these names will generate a compiler error.

Action: Change the names before you rerun the program.

VQA24W THIS WORD '*word*' WAS DISCARDED

Explanation: An unrecognized word was diagnosed and deleted from the text.

Action: If needed, specify a valid word.

VQA25W HAD TO ASSUME THAT THIS '*number*' IS A LEVEL NUMBER

Explanation: An out-of-place literal was diagnosed and assumed to be a level number.

Action: Correct the literal: the period that ends the previous statement may be missing.

VQA26W THIS IS AN ILLEGAL DATA NAME '*dataname*'

Explanation: The displayed *dataname* is a reserved word.

Action: Specify a valid name.

VQA27E THIS ELEMENTARY ITEM'S 'USAGE' '*dataname*' CONFLICTS WITH THE GROUP'S 'USAGE'

Explanation: A conflicting USAGE was found at the elementary group level when propagating a USAGE defined at the group level; the group USAGE was propagated.

Action: Follow your site standards.

VQA28A THE 'RECORD CONTAINS' CLAUSE IS A COMMENT

Explanation: The RECORD CONTAINS clause is unnecessary.

Action: Follow your site standards.

VQA29A THE ABOVE 'REDEFINES' HAS BEEN CORRECTED '*dataname-1* (NOW) REDEFINES *dataname-2*'

Explanation: The preceding REDEFINES error was corrected.

Action: Check the changes.

VQA30A THIS ITEM '*dataname*' IS A GROUP ITEM IN A CONDITION

Explanation: The displayed *dataname* is a group item encountered in an IF statement. Incorrect results can occur if the subordinate items do not agree with the other comparand in CLASS, USAGE, and so on.

Action: Check your program results.

VQA31A THE 'ADD' STATEMENT ABOVE, CONTAINS ITEMS WITH CONFLICTING 'USAGE.' '*dataname-1*' (*subscript*)' USAGE IS DISPLAY '*dataname-2*' USAGE IS COMP-1

Explanation: The displayed data items have different USAGEs and cannot be manipulated.

Action: Convert the data item(s) to a common type.

VQA32A THE 'ADD' STATEMENT ABOVE, USES ITEMS REQUIRING POINT ALIGNMENT. '*dataname-1* (*subscript*) POINT*n1*' '*dataname-2* POINT *n2*'

Explanation: The displayed data items have different decimal point alignments that require padding and cannot be manipulated.

Action: Correct the decimal point alignment(s).

VQA33A THE 'ADD' STATEMENT ABOVE, USES ITEMS WITH DIFFERENT LENGTHS. '*dataname-1* (*subscript*) LENGTH*n1*' '*dataname-2* LENGTH*n2*

Explanation: The displayed data items have different lengths that require padding.

Action: Correct the length(s).

VQA34A THE ADD STATEMENT ABOVE, USES A 'USAGE DISPLAY' ITEM. '*dataname*'

Explanation: The displayed *dataname* is a DISPLAY item and was encountered in an arithmetic or conditional statement.

Action: Respecify the *dataname*.

VQA35A THE 'CORRESPONDING' CLAUSE MAY CAUSE DIFFICULTY IN PROGRAM MAINTENANCE.

Explanation: The use of CORRESPONDING in an ADD, SUBTRACT, or MOVE statement can cause problems in maintenance.

Action: Follow your site standards.

VQA36A 'ON SIZE ERROR' IS EXPENSIVE

Explanation: The use of ON SIZE ERROR is costly in the execution of programs and not recommended in a production environment.

Action: Follow your site standards.

VQA37A '*verb*' IS AN EXPENSIVE STATEMENT

Explanation: Using the displayed *verb* is costly in the execution of programs.

Action: Follow your site standards.

VQA38A THIS ITEM IS AN INEFFICIENT SUBSCRIPT. '*dataname* USAGE DISPLAY'

Explanation: Dataname is a subscript which is either a COMPUTATIONAL-3 or DISPLAY data item requiring type conversion for use.

Action: Follow your site standards.

VQA39A 'GO TO' MAY BE HARMFUL

Explanation: Excessive use of the GO TO statement is not recommended in a structured COBOL environment. Doing so may cause difficulty in program maintenance.

Action: Follow your site standards.

VQA40A FORWARD (DOWN) 'GO TO' LINE NUMBER=*number*, SEQUENCE NUMBER=*number*

Explanation: A GO TO statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure that follows the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

VQA41A BACKWARD (UP!) 'GO TO' LINE NUMBER=*number*, SEQUENCE NUMBER=*number*

Explanation: A GO TO statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure which precedes the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

VQA42A 'GO TO' GENERATED IN ALTER CONVERSION

Explanation: A GO TO statement was generated during the conversion of an ALTER statement.

Action: Follow your site standards.

VQA43A ALTER STATEMENT CONVERTED TO INITIALIZE A CONTROL VARIABLE

Explanation: An ALTER statement has been converted to a switch, a test, and a GO TO statement.

Action: Follow your site standards.

VQA44A THIS '*verb*' STATEMENT, DOES NOT HAVE A 'THRU' CLAUSE, AND IT SHOULD

Explanation: A THRU/THROUGH phrase is required for the SORT INPUT or OUTPUT PROCEDURE, MERGE OUTPUT PROCEDURE, or PERFORM statement displayed.

Action: Follow your site standards.

VQA45A THIS '*statement*' HAS A 'THRU' CLAUSE, AND IT SHOULD NOT

Explanation: The THRU/THROUGH phrase is not needed for the SORT INPUT or OUTPUT PROCEDURE, MERGE OUTPUT PROCEDURE, or PERFORM statement displayed.

Action: Follow your site standards.

VQA46A FORWARD (DOWN) PROCEDURE INVOCATION LINE NUMBER=*number*, SEQUENCE NUMBER=*number*

Explanation: A SORT, MERGE, PERFORM, or EXEC/EXECUTE CICS statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure that follows the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

VQA47A BACKWARD (UP!) PROCEDURE INVOCATION, LINE NUMBER=*number*, SEQUENCE NUMBER=*number*

Explanation: A SORT, MERGE, PERFORM, or EXEC/EXECUTE CICS statement, coded at CA-MetaCOBOL+ LINE NUMBER=*number*, COBOL SEQUENCE NUMBER=*number*, refers to a procedure that precedes the statement. (SEQUENCE NUMBER can be blank if the original input statement was not numbered.)

Action: Follow your site standards.

VQA48A THIS '*verb*' STATEMENT USES THE 'FROM' CLAUSE

Explanation: An I/O statement with a FROM or INTO option was encountered.

Action: Follow your site standards.

VQA49A THIS '*verb*' STATEMENT DOES NOT USE THE 'FROM/INTO' CLAUSE

Explanation: An I/O statement without a FROM or INTO option was encountered.

Action: Follow your site standards.

VQA51A OPERATOR INTERVENTION IN '*statement*' IS A POOR OPERATING PROCEDURE

Explanation: An ACCEPT or DISPLAY statement requesting the use of the system console was encountered.

Action: Follow your site standards.

VQA52A THIS PROCEDURE HAS A PRIORITY NUMBER '*procedure-name* SECTION *nn*'

Explanation: Program segmentation is more costly when running under a VS operating system than the equivalent paging overhead.

Action: Follow your site standards.

VQA53A DEBUGGING VERBS (*debugging-verb*) SHOULD NOT BE USED IN PRODUCTION PROGRAMS

Explanation: An occurrence of one or more of the following verbs was encountered:

READY TRACE
RESET TRACE
EXHIBIT
ON *literal*

Debugging statements should not exist in a production environment.

Action: Follow your site standards.

VQA54E THIS SUBSCRIPT SYNTAX IS INVALID. '*dataname* (*subscript*)'

Explanation: A subscript which is not valid COBOL syntax has been encountered.

Action: Correct the subscript syntax.

VQA55A THIS PROCEDURE MAY BE ACCOMPLISHING A SEQUENTIAL TABLE SEARCH '*procedure-name*'

Explanation: A sequential search of a table may have been encountered.

Action: A SEARCH statement should be specified for sequential table searches.

VQA57E THIS PARAGRAPH NAME IS A DUPLICATE '*procedure-name*'

Explanation: The displayed *procedure-name* is not unique. Numbering cannot make it a unique name because CA-MetaCOBOL+ cannot determine which procedure is relevant.

Action: Change the name of at least one of the procedures in the section or program, then try again.

VQA58E THIS PROCEDURE NAME IS UNDEFINED '*procedure-name*'

Explanation: The displayed *procedure-name* cannot be located.

Action: Correct the error, then try again.

VQA60A THIS NAME IS TOO SHORT '*name*'

Explanation: The displayed '*name*' is shorter in length than the number specified by the $CQA-SHORT control statement.

Action: Follow your site standards.

VQA61A THIS PROCEDURE NAME, '*procedure-name-1*' HAS BEEN TRUNCATED '*proc-name-2*' DUE TO PROCEDURE NUMBERING

Explanation: The displayed *procedure-name-1* is longer than 30 characters when prefixed with a 4-digit sequence number and a hyphen. It has been truncated to *proc-name-2*, which is 30 characters in length.

Action: Follow your site standards.

VQA62A THIS DATA NAME IS UNDEFINED '*dataname*'

Explanation: The displayed *dataname* or Data Division section-name is not defined, and is therefore unavailable for Data Division mapping.

Action: Define *dataname* or remove its reference from the $VQA-MAP control statement.

VQA63A THIS PROCEDURE NAME IS UNDEFINED, '*procedure-name*' BUT MAY BE IN A COPYBOOK

Explanation: Reference was made to an undefined procedure. A warning (VQA63W) is issued when unprocessed COPY statements are also present. An error (VQA63E) is issued when COPY statements are not present or when COPY=ACTIVE is specified and all COPYs have been processed.

Action: To correct the warning, specify COPY=ACTIVE. To correct the error, define the procedure.

VQA64W AT LEAST ONE COPYBOOK IS UNAVAILABLE, RESULTS MAY BE INCORRECT

Explanation: One or more COPY elements were unavailable for processing.

Action: Specify the COPY=ACTIVE or -INC translate-time options.

VQA65A THE FOLLOWING IS AN IBM EXTENSION '*extension*'

Explanation: The displayed *extension* is an IBM extension to the 1974 ANSI COBOL standard. It has been retained to ensure upward portability from the Version 4 compiler, available under non-VS operating systems.

Action: Follow your site standards.

VQA66W A 'NULL' GO TO STATEMENT HAS BEEN DISCARDED

Explanation: A GO TO statement that does not have a destination has been discarded. CA-MetaCOBOL+ assumes that an ALTER statement has replaced the GO TO with an IF statement and a GO TO statement.

Action: Check the conversion to make sure program control is not adversely affected, then rerun.

VQA67A THE DATA DIVISION HAS BEEN COMPLETED TO SUPPORT DATA MAPPING

Explanation: Regardless of the absence of a Procedure Division header, VQA completed the internal tables, thus permitting mapping.

Action: No action required. However, a Procedure Division header can be entered.

VQA68W THIS PROGRAM HAS TOO MANY PROCEDURE NAMES, GIVEN THE '$VQA-SEQN' PARAMETER SPECIFIED (number). PROCEDURE NAME NUMBERING HAS BEEN DISCONTINUED

Explanation: The number of paragraphs and the paragraph numbering increment have caused the renumbering process to overflow.

Action: Specify a smaller value for the $CQA-SEQN parameter.

VQA69E THIS PROCEDURE NAME IS ALSO A DATA NAME '*procedure-name*'

Explanation: The displayed *procedure-name* was also used as a data name; no renumbering will be done.

Action: Change the paragraph or data name to make them unique.

VQA70E MORE THAN 5000 PARAGRAPHS ARE REFERENCED PRIOR TO THEIR DEFINITIONS. PARAGRAPH NUMBERING HAS BEEN DISCONTINUED

Explanation: All of CA-MetaCOBOL+'s OUT-OF-LINE markers (&MARKER) have been utilized. A marker is required for each:

- reference to a paragraph or section which precedes the definition of the paragraph or section
- GO TO paragraph or section
- ALTER statement

Action: Follow your site standards.

VQA71A 'REDEFINES FILLER' WAS NOT CHANGED

Explanation: REDEFINES FILLER is vague.

Action: Provide a name for the entity you want to redefine, then try again.

VQA72A THIS IS AN INVALID LEVEL NUMBER, '*nn*'

Explanation: The displayed level number is invalid. Valid values are 1-49, 66, 77, or 88.

Action: Check your program, make necessary corrections, and rerun.

VQA73E A STRING MACRO CALL ->*string macroname*<- HAS BEEN IGNORED

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL+ Support.

VQA74A A MISSING PERIOD HAS BEEN RESTORED

Explanation: Required punctuation was inserted by CA-MetaCOBOL+

Action: Check your program, and verify the changes.

VQA75E AN 'ALTER' TARGET IS NOT A 'GO TO' PARAGRAPH. LINE NUMBER = *number* SEQUENCE NUMBER = *number*

Explanation: The target of an ALTER statement must be a GO TO statement.

Action: Check your program, make sure the target of the ALTER statement is a GO TO statement, then try again.

VQA77W '$VQA-*name value-1*' IS INVALID. DEFAULT STATUS (*value-2*) UNCHANGED

Explanation: An invalid value was specified for a $VQA-*name* statement. Valid values are Y (YES) or N (NO). The statement has accepted its default.

Action: Check your program logic. If the default is appropriate, try to rerun the program. If the default is not appropriate, change the value before you try to rerun the program.

VQA78E VS COBOL II QUALITY ASSURANCE REQUIRES 'OPTION DIALECT=XO'

Explanation: An invalid value was specified for the CA-MetaCOBOL DIALECT option. The valid value is 'XO'.

Action: Specify the valid value, and try again.

VQA79W '$VQA-RELATIONS *value-1*' IS INVALID. DEFAULT STATUS (*value-2*) UNCHANGED.

Explanation: An invalid value was specified for a $VQA-RELATIONS *name* statement. Valid values are S (SHORT) or L (LONG). The statement has accepted its default.

Action: Check your program logic. If the default is appropriate, try to rerun the program. If the default is not appropriate, change the value before you try to rerun the program.

VQA80E DATADICTIONARY INTERFACE FAILURE. SIGNON ERROR

Explanation: The CA-DATADICTIONARY interface has failed as specified.

Action: Check the JCL, DATAVIEW statement, program-name, DDID, PSTAT and PVER options.

VQA81W '$VQA-SHORT *value*' IS INVALID. DEFAULT STATUS (*nn*) UNCHANGED

Explanation: An invalid value was encountered in a $VQA-SHORT control statement.

Action: The value must be an integral numeric literal less than 30 or 0; correct the $VQA-SHORT control statement.

VQA82W '$VQA-SEQN *value*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED

Explanation: An invalid value was encountered in a $VQA-SEQN control statement.

Action: The value must be an integral numeric literal less than 1000; correct the $VQA-SEQN control statement.

VQA83W 'VQA-MAP' SPECIFIES TOO MANY NAMES. ONLY THE FIRST 10 WILL BE USED

Explanation: More than 10 data structures were entered in the $VQA-MAP control statements(s). Only the first 10 will be mapped.

Action: Correct the $VQA-MAP control statement.

VQA90W '$VQA-INCR *value*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED.

Explanation: An invalid value was encountered in a $VQA-INCR control statement.

Action: The value must be a numeric literal or 0; correct the $VQA-INCR control statement.

VQA92W THIS PROCEDURE DIVISION CONTAINS COPY STATEMENTS. PROCEDURE NAMES IN THE COPYBOOKS HAVE NOT BEEN NUMBERED

Explanation: The COPY=PASSIVE or COPY=IGNORE translate-time option has been specified, preventing the processing of copybook contents. The second clause is issued only when procedure renumbering is being performed.

Action: Specify COPY=ACTIVE.

VQA93E THIS IS A DUPLICATE SECTION NAME. '*section-name*'

Explanation: A section or paragraph name is used more than once.

Action: Section names must be unique; paragraph names must be unique or qualified.

VQA94A THIS IS A NON-UNIQUE PARAGRAPH NAME '*paragraph-name*'

Explanation: The PROCEDURE DIVISION contains a non-unique paragraph name.

Action: Specify a number greater than 0 in the $VQA-SEQN control statement or rename the paragraph name.

VQA95E THIS CICS STATEMENT IS INVALID

Explanation: An incorrectly terminated EXEC/EXECUTE CICS statement was located in the Procedure Division.

Action: Review syntax and correct the statement.

VQA98E $PDX CODE GENERATION ERROR.
STATE=0
SYMBOL=NA

Explanation: An internal error has occurred.

Action: Report this problem to CA-MetaCOBOL+ Support.

VQA99W '$VQA-*clause*-COLUMN-*nn*' IS INVALID. DEFAULT VALUE (*nn*) UNCHANGED.

Explanation: An improper integer was specified with the $VQA-*clause*-COLUMN control statement; columnar alignment is ignored.

Action: Correct the $VQA-clause-COLUMN control statement.

B.3.2 VQA/DVQA Internal Error Diagnostic

The following diagnostic is not numbered. It is issued only when VQA discovers what appears to be an internal logic error. The diagnostic and all supporting documentation should be supplied to your Computer Associates representative for corrective action.

VQA INTERNAL ERROR - FD

Explanation: VQA has discovered its own error in the FD macro.

Action: Contact your Computer Associates representative.

Index