Advantage™ VISION:Results™ for z/OS

Toolkit Reference Guide

r6

Contents

Chapter 4: WebSphere MQ Series Support

Chapter 5: CSV Utility

Chapter 6: PCFILE and PCWRITE

Chapter 7: Library Functions

Chapter 8: APPC and z/OS

Chapter 9: Label Generation

Chapter 10: VISION:Pds

Chapter 11: VISION:Table

Chapter 12: Starting VISION:Report

Chapter 13: Bridge to VISION:Inquiry

Index

Chapter 1: Introduction

This guide comprehensively describes the subroutines and special procedures for Advantage™ VISION:Results™ for z/OS.

Chapter 2: Macro Facility	Describes the MACRO facility, explaining what a macro is and how to use the commands and keywords associated with macros.
Chapter 3: Freeze and Restore Features	Describes the FREEZE and RESTORE features. These features give you the ability to freeze and restore machine instructions (generate object code) produced by VISION:Results for a particular program.
Chapter 4: WebSphere MQ Series Support	Describes the WebSphere MQ Series Message Queue Interfaces (MQIs). These interfaces allow you to customize your program to effectively use the WebSphere MQ features.
Chapter 5: CSV Utility	Describes CSVRSLT and CSVSYST, which are application programmer interfaces that convert a field of any type and length to a comma-separated values (CSV) format.
Chapter 6: PCFILE and PCWRITE	Describes the PCFILE statement and the PCWRITE command. These statements allow data to be written to a sequential data set or a temporary JES for download by Advantage™ VISION:Intraaccess™, or to the Advantage™ VISION:Journey® for Windows VSAM download file. This feature is only available to z/OS VISION:Results users who also subscribe to VISION:Intraaccess or VISION:Journey for Windows.
Chapter 7: Library Functions	Describes the library functions, including time routines, numeric conversions, and date calculations and conversions.
Chapter 8: APPC and z/OS	Describes support of cooperative and distributed applications between z/OS systems using APPC and z/OS, allowing a VISION:Results z/OS program to initiate peer-to-peer communications with a remote VISION:Results z/OS program by using LU 6.2 protocols. APPC and z/OS also allow any program using SAA Common Program Interface Communications (CPI Communications) services on any platform to initiate peer-to-peer communications with a remote z/OS VISION:Results program by using LU 6.2 protocols.

Chapter 9: Label Generation	Describes VISION:Results label generation feature.
Chapter 10: VISION:Pds	Describes the optional subroutine Advantage™ VISION:Pds™ (z/OS). VISION:Pds allows the user to access a z/OS partitioned data set (PDS). The user can selectively extract members and directory entries and return them to VISION:Results for interrogation and printing. You must be licensed to use the VISION:Pds subroutine.
Chapter 11: VISION:Table	Describes the optional subroutine Advantage™ VISION:Table™. VISION:Table gives extensive table handling capability to z/OS and VSE users. You must be licensed to use the VISION:Table subroutine.
Chapter 12: Starting VISION:Report	Describes how to execute Advantage™ VISION:Report™ from VISION:Results. From your VISION:Results program, you can start stand-alone VISION:Report applications that access IMS or CA-IDMS® and DB data bases.
Chapter 13: Bridge to VISION:Inquiry	Describes the Bridge facility that starts the Advantage™ VISION:Inquiry® Batch query facility from VISION:Results. To use the Bridge, you must have VISION:Inquiry r6 (or higher) installed in your data center.

Note: Advantage VISION:Results will be referred to as VISION:Results throughout this guide.

Contacting CA Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support.

Chapter 2: Macro Facility

What Is a Macro?

A macro is composed of two parts: specific immediate requests to VISION:Results (meta-statements) and normal VISION:Results statements.

You can code any VISION:Results statement (except COPY and other macros) inside a macro. The COPY statement and calls to other macros are handled by the VISION:Results macro processor. These statements can both modify the VISION:Results code that follows (creating different versions depending upon the parameters that you supply when you start the macro) and either include or exclude certain sections of VISION:Results code (again depending upon the parameters that you specify).

Why Use a Macro?

A macro can be thought of as a skeleton solution to a problem. A macro is controlled by the parameters that you supply it. This allows you to keep one generic solution around, and then modify it to fit your current needs.

The advantages of using macros include:

- Modify only one piece of code (the macro) when a repetitive change is requested.
- Eliminate the risk of overlooking code interdependencies.
- Make program maintenance easier.

Where Is a Macro Kept?

Under z/OS, macros are kept in a partitioned data set (PDS) identified by the SYSCOPY ddname. Under VSE, the LIBDEF JCL statement points to the source statement library (SSL).

In addition, macros can be kept in AllFusion® CA-Panvalet®, AllFusion® CA-Librarian®, AllFusion® Endevor®, and CONDOR (VSE only) libraries.

How Do You Start a Macro?

To start a macro, specify a percent sign (%) followed by the name of the macro using the syntax:

```
%macroname [PDS | SSL | LIB | PAN | CON | NDV] [!variable symbols!]
```

You can optionally append the type of library to be searched:

- PDS for partitioned data set (default for z/OS)
- LIB for CA-Librarian data sets
- NDV for Endevor elements under z/OS
- PAN for CA-Panvalet data sets
- SSL for source statement library (default for VSE)
- CON for CONDOR under VSE

```
%ACTDATA LIB
```

You can also specify parameters and their associated values. Each parameter starts with a pound sign (#) and is terminated by an equal sign (=). The value that you want to use for that parameter immediately follows the equal sign.

```
%ACTDATA #FORM=SHORT #TYPE=IRS
```

You can assign a non-value (NULL) to a parameter:

```
%ACTDATA #FORM=NULL
```

In addition, you can assign the value NULL to any parameter by not specifying any value. However, the last parameter must have a value specified; you cannot use the default null value.

```
%ACTDATA #FORM= #TYPE=IRS
```

Parameters can be tested with conditional statements, but the test determines if the parameter has a value or not. Parameters that have been set to NULL have no value; parameters set to anything else have a value.

Macro Definition Elements

This section explains each of the statements that can be used inside a macro. It explains the order of precedence when more than one value is given for a parameter, and for how long the setting of a parameter is valid.

$DEFAULT Statement

This statement starts the definition of all parameters that are going to have default values when the macro is executed. If you do not specify other values when you start the macro, these values are in effect. Values specified when you start the macro override these defaults. The $DEFAULT statement must be the first statement in the macro and must be terminated with a $DEND.

```
$DEFAULT
#1=REJECT
#TWO=ACCEPT
#ALPHA=TESTING  #BETA=DEBUG
#STR="THIS IS A STRING VALUE"
$DEND
```

You can code more than one default value on a line, but you cannot code part of a default value on one line and then continue it onto the next line.

Use double quotation marks (“) around strings that contain embedded blanks.

Default variables are local to the macro that defines them. They are not retained when the macro completes processing.

$GDEF Statement

The $GDEF statement defines a global variable and sets a default value for it. The difference between this statement and $DEFAULT is that the value of the global variable is available to the macro that issued the $GDEF and all subsequent macros in the same program, although the symbol is marked inactive (null) after each macro invocation. To use the value of this symbol in subsequent macros, issue another $GDEF with the same variable name and any value. The new value will be ignored (only the first $GDEF sets a value) and the status of the variable will be changed to active.

The format of the $GDEF statement and the rules for usage are the same as the $DEFAULT, except that it begins with $GDEF and ends with $ENDGDEF.

```
$GDEF
#SIZE=SMALL
$ENDGDEF
```

$GSET Statement

The $GSET statement can be used to either modify the value of a previously defined global variable or it can be used to define a new global variable. When a $GSET is issued, the value of the variable is changed regardless of what has happened previously with this parameter. End this statement with $ENDGSET.

```
$GSET
#SIZE=TALL
$ENDGSET
```

$SET Statement

The $SET statement is used to give new values to local variables. It can be used anytime within a macro. The format and restrictions of $GDEF and $GSET apply here also. End this statement with $ENDSET.

This statement is the equivalent of a $GSET for local variables. It acts as the defining command (if the variable is not yet defined) and always resets to the newly specified value (regardless of the previous value).

$IF Statement

The $IF statement tests to see whether or not a variable has been defined, and, if so, whether or not the value is NULL.

If the variable has been defined and its value is not NULL, the code up to the next $IFE command is included in the VISION:Results program. (This can be either more macro variable assignment statements, normal VISION:Results syntax, or VISION:Results syntax with macro variables.)

If the variable has not been defined, or if it has been defined and its value is NULL, the code up to the next $IFE statement is not included.

The $IF statement and the variable that it is testing must be on one line; no other statement or command can be on that line.

The $IFE statement must be on a line by itself.

You cannot nest $IF commands.

$IFVALUE Statement

The $IFVALUE clause can be used to control conditional compilation in a manner similar to that of $IF.

This clause can only be used inside of a macro and has the following format:

```
$IFVALUE variable name comparison operator value
```

or:

```
$IFVALUE variable name unary operator
```

Variable name — is a # symbol followed by 1-10 alphanumeric characters.

Comparison operator — is one of the following: EQ, NE, LT, LE, GT, or GE. They have the same meaning as in normal VISION:Results code.

Value — is a number (with or without an optional leading sign), a character string, or a character string enclosed in single or double quotation marks. The character string can be up to 30 bytes long.

Unary operator — is one of the following: NEGATIVE, POSITIVE, NUMERIC, or ALPHABETIC. They have the same meaning as in normal VISION:Results code.

The $IFVALUE clause produces the same kind of actions as the $IF clause: it either includes or excludes the next section of the macro in the compilation. For example:

```
$IFVALUE #STATS EQ "YES"
     MOVE TEMPSTAT TO OUTFILEB.STAT
$IFE
```

#STATS can be either a local or a global macro symbol. If it currently has the value of YES and is not an inactive global variable, the statement following the $IFVALUE clause is included in the compilation. If it is not defined, has a different value than YES, or is an inactive global variable, the statement does *not* become a part of the compilation.

All restrictions that apply to $IF also apply to the use of $IFVALUE; the nesting of $IF and $IFVALUE clauses is not supported.

$ELSE Statement

The $ELSE clause allows the inclusion of an alternate path in an $IF or an $IFVALUE clause.

```
$IF #STATS
     MOVE TEMPSTAT TO OUTFILEB.STAT
     WRITE OUTFILEA
$ELSE
     MOVE REALSTAT TO OUTFILEA.STAT
     WRITE OUTFILEA
$IFE
```

If the macro preprocessor variable #STATS is defined, not null, and not an inactive global variable, the first two VISION:Results statements are included in the compilation but the last two are not. If #STATS is not defined, null, or an inactive global variable, the first two VISION:Results statements are not part of the compilation but the last two are.

$ELSE must be the only thing on the input line.

$ELSE can be used with either $IF or $IFVALUE clauses. It is an error to use $ELSE without a preceding $IF or $IFVALUE statement.

Parameters Inside of a Macro

All of the above elements would be useful by themselves, but not particularly powerful. The power of a macro comes from its ability to make substitutions when it is started.

You can use the value that you have assigned to a parameter (whether directly when the macro is started or indirectly through the use of a DEFAULT statement) anywhere within a macro by specifying a pound sign (#) followed by the parameter name.

```
MOVE #MYPARM TO FIELD1
```

You can also concatenate the parameter with statements:

```
MOVE #MYPARM1#MYPARM2 TO WHENCE
MOVE #MYPARM|NAME TO SAFETY
MOVE PREFIX#MYPARM TO HOME
```

You can append the value of a parameter to either another parameter or a character string by specifying # followed by the parameter name. (No intervening blanks are allowed.)

Y and Z are two reserved parameters that are useful in a macro.

In z/OS, Y is a one-character field containing an O, and in VSE, Y is a one-character field containing a D.

```
$IFVALUE #Y EQ 'D'
  SORT ARFILE USING #1 #2 WORK 3
```

Z is a two-character field containing the number of times a macro has been used in this program. It contains 01 the first time a macro is used, 02 the second time a macro is used, and so on. When you define a field such as TEMPFLD in a macro, append a #Z to the end of the field name, making it TEMPFLD#Z. Each time you refer to this field in the macro, use the name TEMPFLD#Z. In this way, you can use the same macro more than once in the same program and not define the same field name twice, which would cause an error.

```
WORKAREA
TEMPFLD#Z  6  PD 2
TEMPFLD#Z = TEMPFLD#Z + #INFLD
```

You can append a string value to a parameter by specifying a vertical bar (|) followed by the character string. (No intervening blanks are allowed.)

You can make as many appends as you want; however, you cannot append inside of one of the various set commands (such as, GDEF GSET, SET) or on a $IF statement.

Examples

Example 1 Using $IF

Assume that you have two files that start with the same types of fields, but that the first file has some additional fields that are not present in the second file. Here is a macro that might help:

```
$DEFAULT
  #INCLUDE=NULL
  #LRECL=80
$DEND
FILE #FNAME FB #LRECL #LRECL|0
  #FNAME|NAME    30 CH
  #FNAME|ADDRESS 40 CH
  #FNAME|BALANCE 10 PD
$IF #INCLUDE
  #FNAME|PAYMENT 5 PD
  #FNAME|BRANCH  5 PD
$IFE
```

Figure 1 Example Using $IF

Assume that the macro has been placed in a library and has the name CUSTOMER. If you then invoked it twice with the following commands:

```
%CUSTOMER #FNAME=CHECKING
%CUSTOMER #FNAME=LOAN #INCLUDE=YES #LRECL=90
```

you would then get the following statements in your VISION:Results program:

```
FILE CHECKING FB 80 800
  CHECKINGNAME    30 CH
  CHECKINGADDRESS 40 CH
  CHECKINGBALANCE 10 PD
FILE LOAN FB 90 900
  LOANNAME    30 CH
  LOANADDRESS 40 CH
  LOANBALANCE 10 PD
  LOANPAYMENT  5 PD
  LOANBRANCH   5 PD
```

Figure 2 Example Using $IF

Example 2 Using $SET (Local Command)

Macro MONTHLY1

```
$DEFAULT #1=ACCOUNT #2=TRANS #3="'MONTHLY REPORT'"
$DEND
$IF #SEQ1
  $SET #1=TRANS #2=ACCOUNT $ENDSET
$IFE
CONTROL ACCOUNT
IF ACCOUNT EQ 'NA' 'EO' THRU 'IO' NEXT ELSE REJECT
ENDIF
SORT ARFILE USING #1 #2
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
```

Figure 3 Example Using $SET (Page 1 of 2)

```
ON CHANGE IN ACCOUNT
  LIST SUM BALANCE WITH 2 AFTER AND 2 BEFORE
ON FINAL
  LIST SUM BALANCE WITH 2 BEFORE
  LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
  TALLY AT BALANCE
T1 #3 WITH 2 AFTER
 T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 3 Example Using $SET (Page 2 of 2)

The macro MONTHLY1 is started by %MONTHLY1 and produces the following:

```
SORT ARFILE USING ACCOUNT TRANS
.
.
T1 'MONTHLY REPORT' WITH 2 AFTER
```

If the macro MONTHLY1 is started by %MONTHLY1 #SEQ1=YES, it produces:

```
SORT ARFILE USING TRANS ACCOUNT
.
.
T1 'MONTHLY REPORT' WITH 2 AFTER
```

Example 3 Using $GSET and $GDEF (Global Commands)

Macro FILE2

```
$DEFAULT  #FIELD1=ACCOUNT #FIELD2=TRANS #FIELD3=NAME
          #FIELD4=BALANCE
$DEND
$GSET   #FILENAME=ARFILE $ENDGSET
FILE #FILENAME FB 352  5280
  #FIELD1  2 182
  #FIELD2  7   4
  #FIELD3 25  85
  #FIELD4  5 170 PD 2 A
```

Figure 4 Example Using $GSET

Macro MONTHLY2

```
$DEFAULT #1=ACCOUNT #2=TRANS #3="'MONTHLY REPORT'"
$DEND
$GDEF #FILENAME=FILEX $ENDGDEF
$IF #SEQ2
  $SET #1=TRANS #2=ACCOUNT $ENDSET
$IFE
CONTROL ACCOUNT
IF ACCOUNT EQ 'NA' 'EO' THRU 'IO' NEXT ELSE REJECT
ENDIF
SORT #FILENAME USING #1 #2
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
ON CHANGE IN ACCOUNT
  LIST SUM BALANCE WITH 2 AFTER AND 2 BEFORE
ON FINAL
  LIST SUM BALANCE WITH 2 BEFORE
  LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
  TALLY AT BALANCE
T1 #3 WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 5 Example Using $GDEF

The following sequence:

```
%FILE2
%MONTHLY2
```

produces the following:

```
SORT ARFILE USING ACCOUNT TRANS
T1 'MONTHLY REPORT'
```

Example 4 Using Field and File Variable Symbols

Macro FILE3

```
$DEFAULT  #FIELD1=ACCOUNT #FIELD2=TRANS #FIELD3=NAME
          #FIELD4=BALANCE
$DEND
$GSET #FILENAME=ARFILE $ENDGSET
FILE #FILENAME  FB  352  5280
  #FIELD1  2 182
  #FIELD2  7   4
  #FIELD3 25  85
  #FIELD4  5 170 PD 2 A
WORKAREA
  DEPT 5 PD ('HEADER DEPT')
```

Figure 6 Example Using Field and File Variable Symbols

Macro MONTHLY3

```
$DEFAULT #1=ACCOUNT #2=TRANS #3="'MONTHLY REPORT'"
$DEND

$GSET #FILENAME=ARFILE $ENDGSET
WORKAREA
  AVG1 5 NU ('HEADER'AVG1')
$IF #KEY1
  #KEY1|AVG  #L'#KEY1  #T'#KEY1  ('#H'#KEY1')
$IFE
$IF #SEQ2
  $SET #1=TRANS #2=ACCOUNT $ENDSET
$IFE
CONTROL ACCOUNT
IF ACCOUNT EQ 'NA' 'EO' THRU 'IO' NEXT
  ELSE REJECT ENDIF
SORT #FILENAME USING #1  #2
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
ON CHANGE IN ACCOUNT
  #KEY1|AVG=SUM BALANCE/TALLY
  LIST 'AVERAGE' AT ACCOUNT #KEY1|AVG WITH 2 AFTER
  SUM BALANCE WITH 2 AFTER AND 2 BEFORE
ON FINAL
   LIST SUM BALANCE WITH 2 BEFORE
  LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
  TALLY AT BALANCE
T1 #3 WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 7 Example Using Field and File Variable Symbols

The macro MONTHLY3 is started by %MONTHLY3 #KEY1=DEPT. It checks the #KEY1 variable symbol and, if this is defined and not null, the following statement is generated:

```
DEPTAVG 5 PD ('HEADER DEPT')
```

In addition, the following field variable symbols are checked:

#L'	Length of the field.
#T'	Data type of the field.
#H'	Header for the field.

Advanced Techniques

There are three other groups of commands that can be issued inside a macro that are useful in specialized situations.

PRINTGEN and PRINTNOGEN

The default is to have all of the statements within the macro (with the exception of $IF and $IFE) printed with the rest of your VISION:Results code. You can use the command PRINTNOGEN to turn off printing of this code. You can use the command PRINTGEN to turn printing back on.

Attributes

The attributes of fields and files can be accessed within a macro by specifying #, a one-letter key code, a single quotation mark ('), and the name of the file or field.

#C'filename	VISION:Results' internal alias count field name for the file.
#D'dataname	Number of decimals in the field.
#E'dataname	Edit code for the field.
#H'dataname	Column heading for the field.
#L'dataname	Length of the VISION:Results internal alias field.
#R'filename	Record length field name for the file.
#S'filename	VISION:Results internal alias status field name for the file.
#T'dataname	Data type of the field.
#P'dataname	Starting position of the field.

Debugging

To assist you in debugging a complicated macro, the $PRINT command determines the value of a parameter while the macro is being processed.

```
$PRINT
```

The $PRINT command lists all known parameters (global and local), their length, how they were set (as defaults, set inside the macro, or as overrides when the macro was started), and what their current value is.

Chapter 3: Freeze and Restore Features

Freeze Feature

The freeze module feature can be used to save the machine instructions (generated object code) produced by VISION:Results for a particular program. These instructions can be link edited to a load library (z/OS) or a phase and core image library (VSE) in the same manner as output produced by compilers and assemblers. Frequently run programs that have been checked out can be frozen so that the program does not have to be validated and compiled each time prior to execution. This is beneficial for several reasons:

- Decrease in execution time.
- Better control over programs.

The freeze module produced by a VISION:Results program is not a stand-alone program, but a subprogram. VISION:Results is still required to execute a frozen module. Usually, only the coding and data which is constant are frozen. This helps to conserve library space.

Procedure

A VISION:Results program should be run in the normal manner until it is completely checked out and debugged. When a program is to be frozen, the following OPTION statement should be placed at the beginning of the VISION:Results program:

```
OPTION FREEZE modulename [DATA 'literal']
```

Where:

- modulename is the 1- to 8-character alphanumeric name to be assigned in this program. Any data input following the FIN should be removed.
- literal is any 38 bytes of information (enclosed in quotation marks) that is inserted during the execution of a restore of this module name.

Freezing and Restoring a VISION:Results Program

When a VISION:Results program is frozen, one or two object programs per request can be generated. The generation of the second object program depends on whether the letter writing, linear regression, random and interval selection, scatter diagrams, and trend line functions are used within a request. The second object program is generated only when one or more of these functions is used in a request.

The name assigned to each of the second object programs is derived by appending a dollar sign ($) and number to the freeze name in the OPTION statement. Because an object program name can be up to 8 characters, the freeze name can be truncated to allow for the extra dollar sign and number. The number reflects the ordinal position of the request; that is, the primary request is assigned $1, the second request is assigned $2, the 99th request is assigned $99. For example, in the following program, object programs with the names TSTPROG and TSTPRO$1 are generated for the primary request, TSTPRO02 and TSTPRO$2 for REPORT2, TSTPRO03 for REPORT3, and TSTPRO04 and TSTPRO$4 for REPORT4. The TSTPRO$3 object program is not generated because the third request does not include any of the functions mentioned above.

```
OPTION FREEZE TSTPROG
FILE ARFILE RETAIN
FILE OARFILE OUTPUT FROM ARFILE
  SAMPLE 01 RANDOM 200 73
IF SAMPLING 01 ACCEPT ELSE REJECT ENDIF
REPORT2
USE ARFILE
SAMPLE 02 INTERVAL 10 20
IF SAMPLING 02 HEXPRINT ARFILE
  ELSE REJECT ENDIF

REPORT3
USE ARFILE
  .
  .
  .
REPORT6
USE ARFILE
LINEAR 02 ............
  .
  .
  .
```

Figure 8 Example Freezing a VISION:Results Program

To restore the above program, the following statement would be submitted:

```
OPTION RESTORE TSTPROG 4
```

where 4 is the number of requests to be restored.

JCL Requirements for a Freeze Run

The JCL required for a freeze run is minimal because VISION:Results does not proceed to the execution phase during an option freeze run. All JCL related to the actual running of the program should be omitted.

z/OS Freeze JCL Requirements

The following JCL is required for a z/OS freeze run:

- JOB statement.
- JOBLIB or STEPLIB (optional). To locate the VISION:Results program.
- EXEC statement:

```
//S1 EXEC PGM=DYL280
```

- SYSPRINT DD statement. For statement listing and error messages.
- SYS280FZ DD statement. Identifies the output data set that is to hold the freeze object module produced by VISION:Results. This data set is the primary input to the link edit run and can be on disk, tape, and so forth. The default attributes are fixed record format, 80 characters-per-record, and 80 characters-per-block. The data set can be written as a fixed block file with 80 characters-per-record. For multiple request programs, SYS280FZ should be assigned with a disposition of MOD.

You can override the ddname SYS280FZ as a default by using the DYLINSTL macro parameter FREEZDD or by using the OPTION FREEZDD statement in your program.

- SYS004 — VISION:Results output and input workfile.
- AUDEPF — VISION:Results second work file.
- AUDCBF — VISION:Results third work file.
- AUDWORK — VISION:Results fourth work file.
- AUDPRINT — Additional statement listings and statistics.
- SYSIN — Identifies the input data set containing the VISION:Results statements.

VSE Freeze JCL Requirements

A VSE freeze run results in an object module being written to SYSPCH which can be assigned to disk or tape. The object module is written in fixed format records, 80 characters per record, and 80 characters per block. The format of the output is:

```
// OPTION CATAL
PHASE phasename,+0
 INCLUDE
  - object module -
/*
// EXEC LNKEDT
```

Disk and tape output contain only the PHASE and INCLUDE statements and the object module.

JCL similar to the following example is suggested when freezing a VISION:Results program on a VSE system:

```
// JOB DYLFREEZ
// DLBL IJSYSPH,'FREEZE.OBJ',0
// EXTENT SYSPCH, AND SO ON
 ASSGN SYSPCH,X'cuu'
// ASSGN SYS008,X'cuu'
// DLBL SYS008,'VISION:RESULTS WORK FILE2',0
// EXTENT SYS008,...extent information (20 tracks)
// ASSGN SYS009,X'cuu'
// DLBL SYS009,'VISION:RESULTS LETTERS.SYSLST',0
// EXTENT SYS009,...extent information (20 tracks)
// ASSGN SYS010,X'cuu'
// DLBL SYS010,'VISION:RESULTS WORK FILE3',0
// EXTENT SYS010,...extent information (20 tracks)
// ASSGN SYS011,X'cuu'
// DLBL SYS011,'VISION:RESULTS WORK FILE4',0
// EXTENT SYS011,...extent information (20 tracks)
// EXEC DYL280,SIZE=250K
OPTION FREEZE MYPROG
  (VISION:Results program statements go here)
/*
/&
 CLOSE SYSPCH,X'cuu' (revert to normal assignment)
```

Figure 9 Example Freezing a VISION:Results Program on a VSE System

If IJSYS04 (the VISION:Results compiler work file) is not a standard label, you must also supply JCL for it (for more information about operating characteristics, see the *Advantage VISION:Results for z/OS Installation Guide*).

Link Editing a Frozen Module

The VISION:Results run that creates the freeze module validates the VISION:Results statements and compiles and produces an object module; the job does not proceed to execution. The resulting object module must be link edited. This is done in the standard manner for your system.

z/OS

- A NAME statement is produced following the object module, in the format:

  ```
  NAME modulename(R)
  ```

  where modulename is the value supplied immediately following the Freeze keyword in the OPTION statement.
- The SYSLIN DD statement in the link edit step should point to the same file that was output from the freeze step on SYS280FZ.

VSE

- If you output the object module to disk (see the Freeze example above), the following JCL links the object module into the phase and core image library:

```
// JOB LINK
// OPTION CATAL
// DLBL IJSYSIN,'FREEZE.OBJ',0
// EXTENT SYSIPT, AND SO ON
 ASSGN SYSIPT,X'cuu'
 INCLUDE
// EXEC LNKEDT
/&
 CLOSE SYSIPT,X'cuu' (revert to normal assignment)
```

Figure 10 Example Linking the Object Module Into the Image Library

- The link edit step should catalog the program into the same phase and core image library that VISION:Results is in or into a library that is searched during any VISION:Results run.

Restore Feature

The execution of a frozen module requires that a VISION:Results run be made invoking the frozen module. The number of reports to be generated can also be specified by including that number immediately after the module name. To execute a frozen module, code:

```
OPTION RESTORE modulename nn [[RESTART nnnn] [DELIM [C'c' | X'xx']]
```

where the module name is the 1- to 8-character alphanumeric name assigned to the program in the option freeze run.

The n is where you put a number between 1 and 99 to specify the number of reports (in a multiple report program) to be generated. The generated reports always begin with the first report of the multiple report program, unless specified otherwise.

Restart n (a number from 2 to 99) specifies the report number from which to begin report generation.

DELIM specifies the delimit character to be used when generating the report. REPORT DELIM must be also be specified on the FREEZE program in order to be effective on the RESTORE.

In the following example, MODRPT 3 specifies that three reports be generated and Restart 2 specifies generation should begin with the second report. Thus the second, third, and fourth reports are generated. (For additional information and examples, see Sample VISION:Results Freeze and Execute Frozen Runs on page 29.)

```
OPTION RESTORE MODRPT 3  RESTART 2
```

All other VISION:Results statements are invalid except comments and the following keywords:

STATEOFF	FIN
OPTION DYL4YEAR	OPTION NODYL4YEAR
OPTION CURRENCY	OPTION PRINTEP
NOPRINTEP	REPORT
OPTION $$PCB	OPTION DATA literal
OPTION DISK type	OPTION SYSnnn filename
OPTION NOTOTAL	OPTION PRINTERROR
OPTION RESTART nn	OPTION DELIM [C'c' | X'xx']
OPTION OPTLIST	OPTION NOOPLIST
OPTION COB2NR	OPTION NOCOB2NR
OPTION LE	OPTION NOLE

Note: OPTION LE is only effective for programs frozen under VISION:Results 6.0. Programs frozen under earlier releases must be refrozen using Release 6.0 for OPTION LE to be effective on the RESTORE.

For VSE users, the following keywords are also valid:

OPTION VSEDISK	OPTION NOVSEDISK
OPTION VSETAPE	OPTION NOVSETAPE
OPTION DISK SYSnnn	OPTION VSEALL
OPTION NOVSEALL	

- Input data, if any, must follow the FIN statement on the RESTORE.
- If you specify OPTION DATA, it must have been included in the OPTION statement when the program was frozen; however, you can specify different data if needed. In a multiple report restore run (see Sample VISION:Results Freeze and Execute Frozen Runs on page 29), each report can be restored individually and different data can be passed to each report through the OPTION statement.

OPTION and DYLINSTL Parameter Precedence

The following parameters have their settings determined by how they are used on the RESTORE. OPTION parameters specified on the RESTORE take precedence over the DYLINSTL parameter:

OPTION EXCELPAT/NOEXCELPAT and DYLINSTL EXCLPAT

OPTION LE/NOLE and DYLINSTL LE

The following parameters have their settings determined on the FREEZE and ignore the setting during the RESTORE:

OPTION NUMCHAR/NONUMCHAR and DYLINSTL NUMCHAR

DYLINSTL RETCODE

OPTION NUMPD/NONUMPD and DYLINSTL NUMPD

OPTION ZDIVAB/ZDIVORG/ZDIVRC and DYLINSTL ZDIVAB

REPORT DELIM and DYLINSTL DELIM

OPTION TIMECOLON/TIMEDOT and DYLINSTL TIMESEP

If set during either the FREEZE or RESTORE, either OPTION PRINTERROR or DYLINSTL OPTPRER will be used during the RESTORE.

JCL Requirements for Executing a Frozen Module

The JCL used to execute a frozen module is not different than any other VISION:Results run. The program to be executed remains VISION:Results.

z/OS

- If necessary, include a JOBLIB or STEPLIB statement to locate the frozen module if it is not in an automatically searched load library.
- The UA= subparameter of the z/OS PARM in an EXEC statement can be used to supply values to the DYLPARM reserved word.
- When executing a frozen module you need the JCL for the product under which your program was frozen. For example, a module frozen under VISION:Sixty™ but being restored under VISION:Results requires a SYS260R statement, not SYS280R statement.

VSE

- Be sure that the work file (IJSYS04) is assigned (for more information about operating characteristics, see the *Advantage VISION:Results for z/OS Installation Guide*); it is required for all VISION:Results runs.

Freeze Considerations

When the FILE statements are frozen they cannot be altered at runtime. For z/OS systems and if beneficial, omit coding FILE statement attributes that are supplied in the JCL (such as KEYLOC and BLKSIZE). The only exception is if the file is being sorted, in which case, the attributes (at least record format and record length) are required in the FILE statement.

For more information about the FILE command, see the *Advantage VISION:Results for z/OS Reference Guide*.

Freezing Multiple Reports

A multiple report run can be frozen in the same manner as a single request. Specify an OPTION Freeze module name statement prior to the first request. VISION:Results compiles all requests and produces an object module and link edits control statements for each request. For example, if you have three requests, three object modules are produced in z/OS on SYS280FZ (or the ddname specified on the OPTION or DYLINSTL parameter, FREEZDD) or SYSPCH (VSE). The output from the freeze run must be link edited into a load library (z/OS) or a phase and core image library (VSE) in the same manner as output produced by compilers, before a restore run is attempted.

The first request is given the name specified in the OPTION statement. Subsequent requests have a unique name generated. The name is generated as follows:

- For the 2nd through the 99th requests, a two-digit suffix (02-99) is added to the name specified in the OPTION statement. If the specified name is seven or eight characters long, it is truncated to six characters before adding the suffix.
- For the 100th through the 999th requests, a three-digit suffix (100-999) is added to the name specified in the OPTION statement. If the specified name is six, seven, or eight characters long, it is truncated to five characters before adding the suffix.

JCL Considerations for Freezing Multiple Reports

z/OS

When freezing a VISION:Results multiple report run, you must be sure to specify MOD for the disposition of SYS280FZ. This avoids having each object module being written on top of the previous one.

When specifying MOD, do not specify VOL=SER= on your DD statement. If you do, the system assumes that the data set exists and when VISION:Results attempts to open it, a system 213-04 error occurs.

The following is an appropriate DD statement for SYS280FZ:

```
//SYS280FZ DD DSN=&&DYLOBJ,UNIT=SYSDA,
//            SPACE=(TRK,(3,1)),DISP=(MOD,PASS)
```

A second step should specify this data set as input to the linkage editor (SYSLIN).

VSE

The JCL described in VSE Freeze JCL Requirements on page 23 also works for freezing a multiple report request.

Restoring Multiple Reports

You can restore each multiple report individually by specifying:

```
OPTION RESTORE modulename
```

To execute the entire run, specify the name given in the freeze run, followed by the number of requests to be restored. VISION:Results restores and executes each request in turn. VISION:Results uses the same rule in developing module names as it did when freezing the multiple reports.

The following statement causes the frozen program MREPTS to be run with the associated frozen modules MREPTS02, MREPTS03, MREPTS04, and MREPTS05:

```
OPTION RESTORE MREPTS 5
```

Any modules that are being restored must be in a load library. For z/OS, it must be in a library supplied by a JOBLIB or STEPLIB statement or be in an automatically searched library. For VSE, the module must be in the system phase and core image library or in a private phase and core image library and the appropriate JCL included.

Restart and Restore Execution

Restart can be specified on a VISION:Results restore run.

```
OPTION RESTORE MREPTS 5 RESTART 3
```

The number following the name (MREPTS in this case) is the total number of requests in the program; here it is 5. The number following the Restart keyword is the request to commence processing at; here 3 is specified. This results in MREPTS03, MREPTS04, and MREPTS05 being executed.

Sample VISION:Results Freeze and Execute Frozen Runs

```
//FREEZE JOB
//STEP1 EXEC PGM=DYL280
//SYSPRINT DD SYSOUT=A
//SYS280FZ DD DSN=&&DYLOBJ,UNIT=SYSDA,
//            SPACE=(TRK,(5,5)),DISP=(MOD,PASS)
//SYS004   DD UNIT=SYSDA,SPACE=(TRK,(5,5))
```

Figure 11 z/OS Freeze and Link Edit

```
//AUDPRINT DD SYSOUT=A
//AUDWORK  DD UNIT=SYSDA,SPACE=(TRK,(10,5))
//AUDEPF   DD UNIT=SYSDA,SPACE=(TRK,(10,5)),
//            DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)
//AUDCBF   DD UNIT=SYSDA,SPACE=(TRK,(10,5)),
//            DCB=BLKSIZE=1000
//SYSIN    DD *
OPTION FREEZE MYPROG
  VISION:Results statements
/*
//STEP2 EXEC PGM=IEWL,PARM=NCAL
//SYSPRINT DD SYSOUT=A
//SYSLMOD  DD DSN=your.loadlib,DISP=OLD
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(3,1))
//SYSLIN   DD DSN=&&DYLOBJ,DISP=(OLD,DELETE)
//
```

Figure 11 z/OS Freeze and Link Edit

```
//RUNFZ JOB
//JOBLIB   DD DSN=your.loadlib,DISP=SHR
 //STEP1 EXEC PGM=DYL280
//SYSPRINT DD SYSOUT=A
//FILEIN   DD DSN=INFILE,-
//FILEOUT  DD DSN=OUTFILE,-
//SYS280R  DD SYSOUT=A
//SYS004   DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//AUDPRINT DD SYSOUT=A
//AUDWORK  DD UNIT=SYSDA,SPACE=(TRK,(10,5))
//AUDEPF   DD UNIT=SYSDA,SPACE=(TRK,(10,5)),
//            DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)
//AUDCBF   DD UNIT=SYSDA,SPACE=(TRK,(10,5)),
//            DCB=BLKSIZE=1000
//SYSIN    DD *
OPTION RESTORE MYPROG
FIN
  input data here, if required
/*
```

Figure 12 z/OS Execute Frozen Module

```
// JOB DYLFREEZ
// DLBL IJSYSPH,'FREEZE.OBJ',0
// EXTENT SYSPCH, AND SO ON
 ASSGN SYSPCH,X'cuu'
// ASSGN SYS008,X'cuu'
// DLBL SYS008,'VISION:RESULTS WORK FILE2',0
// EXTENT SYS008,...extent information (20 tracks)
// ASSGN SYS009,X'cuu'
// DLBL SYS009,'VISION:RESULTS LETTERS.SYSLST',0
// EXTENT SYS009,...extent information (100 tracks)
// ASSGN SYS010,X'cuu'
// DLBL SYS010,'VISION:RESULTS WORK FILE3',0
// EXTENT SYS010,...extent information (20 tracks)
// ASSGN SYS011,X'cuu'
// DLBL SYS011,'VISION:RESULTS WORK FILE4',0
// EXTENT SYS011,...extent information (20 tracks)
// EXEC DYL280,SIZE=250K
OPTION FREEZE MYPROG
 VISION:Results statements
/*
/&
 CLOSE SYSPCH,X'cuu' (revert to normal assignment)
```

Figure 13 VSE Freeze

If IJSYS04 (the VISION:Results compiler work file) is not a standard label, you must also supply JCL for it (for more information about operating characteristics, see the *Advantage VISION:Results for z/OS Installation Guide*).

```
// JOB LINK
// OPTION CATAL
// DLBL IJSYSIN,'FREEZE.OBJ',0
// EXTENT SYSIPT, AND SO ON
 ASSGN SYSIPT,X'cuu'
 INCLUDE
// EXEC LNKEDT
/&
 CLOSE SYSIPT,X'cuu' (revert to normal assignment)
```

Figure 14 VSE Link Edit

The link edit step should catalog the program into the same PHASE and core image library that VISION:Results is in or into a library that is searched during any VISION:Results run.

```
// EXECUTE JOB LOADNGO
// ASSGN SYS008,X'cuu'
// DLBL SYS008,'VISION:RESULTS WORK FILE2',0
// EXTENT SYS008,...extent information (20 tracks)
// ASSGN SYS009,X'cuu'
// DLBL SYS009,'VISION:RESULTS LETTERS.SYSLST',0
// EXTENT SYS009,...extent information (100 tracks)
// ASSGN SYS010,X'cuu'
// DLBL SYS010,'VISION:RESULTS WORK FILE3',0
// EXTENT SYS010,...extent information (20 tracks)
// ASSGN SYS011,X'cuu'
// DLBL SYS011,'VISION:RESULTS WORK FILE4',0
// EXTENT SYS011,...extent information (20 tracks)
// EXEC DYL280,SIZE=250K
OPTION RESTORE MYPROG
FIN
  input data, if required
/*
/&
```

Figure 15 VSE Execute Frozen Module

Be sure that the work file (IJSYS04) is assigned (for more information about operating characteristics, see the *Advantage VISION:Results for z/OS Installation Guide*). It is required for all VISION:Results runs.

DYFREZ Subroutine (z/OS Only)

DYFREZ is a program that executes any frozen VISION:Results program. The name of the frozen program is supplied to DYFREZ through the PARM parameter on the EXEC JCL statement. This allows the name of the frozen program to be a variable parameter within a JCL procedure.

DYFREZ uses information from the PARM parameter to create a VISION:Results OPTION Restore statement in the SYSIN data set. Then it transfers control to VISION:Results to execute the frozen program.

To Run DYFREZ

All JCL requirements are the same as for any VISION:Results program, with the following exceptions:

- The EXECUTE statement has the following parameters:

  ```
  //STEP01 EXEC PGM=DYFREZ,REGION=nnnK,PARM='X,Y,Z,UA=U'
  ```

  X = 1-8 bytes, frozen VISION:Results program name.

  Y = 1-2 bytes, the total number of requests.

  Z = 1-2 bytes, number of the request at which to begin processing.

  U = 1-60 bytes, variable data passed to DYLPARM.

- The STEPLIB statement is required:

  ```
  //STEPLIB DD DSNAME=XXXLIB,DISP=SHR (VISION:Results load modules)
  //        DD DSNAME=YYYLIB,DISP=SHR (frozen VISION:Results programs)
  //        DD DSNAME=ZZZLIB,DISP=SHR (DYFREZ load module)
  ```

  All routines can be either linked into the same library or concatenated, as shown above.

- The SYSIN statement:

  ```
  //SYSIN DD UNIT=DISK,SPACE=(TRK,(1,1))
  ```

One restriction must be imposed; instream data cannot come in through SYSIN. Other ddnames must be used, and should be described in the file statements in the VISION:Results frozen program.

DYFREZ Examples

Example 1 Calls a frozen VISION:Results program, using instream data from INFILE file, and puts a literal in DYLPARM.

```
//S001     EXEC PGM=DYFREZ,PARM='TSTRUN,2,UA=END=XXX'
//STEPLIB  DD DSN=DYL.DYL280.LOAD,DISP=SHR
//         DD DSN=DYL.PROD.LIB,DISP=SHR
//SYSIN    DD UNIT=SYSDA,SPACE=(TRK,(1,1)),
//         DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYS280R  DD SYSOUT=A
//SYS004   DD UNIT=SYSDA,SPACE=(TRK,(10,10))
//SYSPRINT DD SYSOUT=A
//INFILE   DD *
AAA TEST REC #1
BBB TEST REC #2
CCC TEST REC #3
DDD TEST REC #4
EEE TEST REC #5
/*
```

Figure 16 Example Calls a Frozen VISION:Results Program

Calls for frozen VISION:Results program TSTRUN. Input is from instream data file (INFILE). Calls for both requests in the frozen VISION:Results program TSTRUN and puts END TEST in DYLPARM. INFILE must be described in a file statement in TSTRUN.

Example 2 Calls a frozen VISION:Results program with ten requests and starts processing at request five.

```
//S001     EXEC PGM=DYFREZ,PARM='TSTRUN,10,5'
//STEPLIB  DD DSN=DYL.DYL280.LOAD,DISP=SHR
//         DD DSN=DYL PROD.LIB,DISP=SHR
//SYSIN    DD UNIT=SYSDA,SPACE=(TRK,(1,1)),
//         DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYS280R  DD SYSOUT=A
//SYS004   DD UNIT=SYSDA,SPACE=(TRK,(10,10))
//SYSPRINT DD SYSOUT=A
//SYS280A  DD DSN=DYL.TEST.DATA,DISP=SHR
/*
```

Figure 17 Example Calls a Frozen VISION:Results Program

Calls for six of the ten requests in the frozen VISION:Results program TSTRUN to execute, starting at request number five.

DYFREZ Subroutine (VSE Only)

DYFREZ is a program that executes any frozen VISION:Results program. The name of the frozen program is supplied to DYFREZ by using the PARM parameter on the EXEC JCL statement. This allows the name of the frozen program to be a variable parameter within a JCL procedure.

DYFREZ uses information from the PARM parameter to create an VISION:Results OPTION Restore statement. Then it transfers control to VISION:Results to execute the frozen program.

To Run DYFREZ

All JCL requirements are the same as for any VISION:Results program with the following exceptions:

- The EXECUTE JCL statement has the following parameters:

  ```
  // EXEC DYFREZ,PARM='X,Y,Z'
  ```

  X = 1-8 bytes, frozen VISION:Results program name.

  Y = 1-2 bytes, the total number of requests.

  Z = 1-2 bytes, number of the request at which to begin processing.

  The first parameter field (X) must always be specified; the others are optional.

- The frozen module should be linked into the same PHASE library as VISION:Results; if it is not, the library it is in should be included in the search chain of a LIBDEF statement as follows:

  ```
  // DLBL DYL282,'YOUR VISION:RESULTS LIBRARY'
  // DLBL DYFREZ,'LIBRARY WHERE DYFREZ RESIDES'
  // LIBDEF PHASE,SEARCH=(DYFREZ.PHASE,DYL282.PHASE)
  ```

DYFREZ Examples

Example 1 Executes a VISION:Results program named TSTFREZ1 which was previously frozen and linked into the VISION:Results PHASE library.

```
// JOB DYFREZ1
// ASSGN SYS004,X'cuu'
// DLBL IJSYS04,'VISION:RESULTS WORK FILE1',0
// EXTENT SYS004,...extent information
// ASSGN SYS008,X'cuu'
// DLBL SYS008,'VISION:RESULTS WORK FILE2',0
// EXTENT SYS008,...extent information    (20 TRACKS)
// ASSGN SYS009,X'cuu'
// DLBL SYS009,'VISION:RESULTS LETTERS.SYSLST',0
// EXTENT SYS009,...extent information    (100 TRACKS)
// ASSGN SYS010,X'cuu'
// DLBL SYS010,'VISION:RESULTS WORK FILE3',0
// EXTENT SYS010,...extent information    (20 TRACKS)
// ASSGN SYS011,X'cuu'
// DLBL SYS011,'VISION:RESULTS WORK FILE4',0
// EXTENT SYS011,...extent information    (20 TRACKS)
// ASSGN SYS012,X'cuu'
// DLBL ARFILE,'INPUT.FILE'
// EXTENT SYS012,DOS001
// DLBL DYL282,'MY.DYL280II.LIBRARY'
// EXTENT SYS022,DOS001
// LIBDEF PHASE,SEARCH=DYL282.PHASE
// EXEC DYFREZ,PARM='TSTFREZ1'
/*
```

Figure 18 DYFREZ Example

Example 2 Executes a frozen multiple request VISION:Results program with six requests and begins processing at the third request.

```
// JOB DYFREZ2
// ASSGN SYS004,X'cuu'
// DLBL IJSYS04,'VISION:RESULTS WORK FILE1',0
// EXTENT SYS004,...extent information
// ASSGN SYS008,X'cuu'
// DLBL SYS008,'VISION:RESULTS WORK FILE2',0
// EXTENT SYS008,...extent information    (20 TRACKS)
// ASSGN SYS009,X'cuu'
// DLBL SYS009,'VISION:RESULTS LETTERS.SYSLST',0
// EXTENT SYS009,...extent information    (100 TRACKS)
// ASSGN SYS010,X'cuu'
// DLBL SYS010,'VISION:RESULTS WORK FILE3',0
// EXTENT SYS010,...extent information    (20 TRACKS)
// ASSGN SYS011,X'cuu'
// DLBL SYS011,'VISION:RESULTS WORK FILE4',0
// EXTENT SYS011,...extent information    (20 TRACKS)
// ASSGN SYS012,X'cuu'
// DLBL ARFILE,'INPUT.FILE'
// EXTENT SYS012,DOS001
// ASSGN SYS022,DISK,VOL=DOS001,SHR
// DLBL DYL282,'MY.DYL280II.LIBRARY'
// EXTENT SYS022,DOS001
// ASSGN SYS023,DISK,VOL=DOS001,SHR
// DLBL DYFREZ,'MY.FROZEN.PROGRAMS'
// EXTENT SYS023,DOS001
// LIBDEF PHASE,SEARCH=(DYFREZ.PHASE,DYL282.PHASE)
// EXEC DYFREZ,PARM='TSTFREZ2,6,3'
/*
```

Figure 19 DYFREZ Example

The frozen program in the above example was linked to a library other than the VISION:Results PHASE library.

Chapter 4: WebSphere MQ Series Support

WebSphere Message Queue (MQ) Series allows programs to communicate with one another across a network of unlike components using a consistent application programming interface.

In message queuing, a *message* is a collection of data sent by one program and intended for another program. *Queuing* is the mechanism by which messages are held until an application is ready to process them. The *message queue* is the named destination to which messages can be sent. Messages accumulate on queues until they are retrieved by programs that service those queues.

Message queuing is a style of program-to-program communication in which each program from an application suite is designed to perform a well-defined, self-contained function in response to a specific request. For communication to occur, a program must put a message on a predefined queue. The other program retrieves the message from the queue, and processes the information contained in the message.

A *queue manager* is a system program that provides queuing services to applications. It provides an application programming interface that allows programs to put messages on, and get messages from, queues.

Using MQIs

VISION:Results provides an easy-to-use interface for defining and identifying message queues, interrogating message status, or adding/retrieving messages from WebSphere MQ. The Message Queue Interfaces (MQIs) provided by VISION:Results enable you to customize your program code to efficiently and effectively use the WebSphere MQ features.

Prerequisites

Before you get started using the VISION:Results MQIs, you should have:

- A WebSphere MQ environment properly established and administered at your site.
- An understanding of the WebSphere MQ nomenclature.
- Knowledge of programming found in *WebSphere MQ Programming Guide, WebSphere MQ Programming Reference Manuals,* and *WebSphere MQ for z/OS Messages and Codes* manual.
- Availability of WebSphere MQ COBOL data definitions for mapping the various structures. The COBOL COPY data structures distributed with z/OS for MQ support should be used for mapping purposes.

Coding COBOL COPY Data Structures

As previously mentioned, the COBOL COPY data structures distributed with z/OS for MQ support should be used for mapping purposes. However, because the z/OS MQSeries COBOL COPY layouts do not start with 01 level data structure, each COBOL COPY data structure should be coded with its own WORKAREA, as follows:

```
WORKAREA
COPY CMQWORK COBOL
WORKAREA
COPY CMQCNOV COBOL
WORKAREA
COPY CMQMDV COBOL
WORKAREA
COPY CMQGMOV COBOL
WORKAREA
COPY CMQODV COBOL
WORKAREA
COPY CMQPMOV COBOL
```

CMQWORK COPY Data Structure

The WebSphere MQ COBOL COPY data structure CMQV cannot be copied into a VISION:Results program because it is already included in the COBOL COPY data structure as CMQWORK. You must use the CMQWORK COPY data structure with VISION:Results. If you include the CMQV COPY data structure, then VISION:Results will terminate with an S0C1 abend because character limitations have been exceeded.

The MQC1_NEW_SESSION field in CMQWORK has been modified to accommodate the VISION:Results limitation of a maximum of 9 hexadecimal characters.

It has been modified from:

```
10 MQCI-NEW-SESSION PIC X(24) VALUE
                     X'414D51214E45575F53455353494F4E5F434F5252454C4944'
```

to the following:

```
10 MQCI-NEW-SESSION.
   15 MQCI-NEW-SESSION1  PIC X(9) VALUE
         X'414D51214E45575F53'.
   15 MQCI-NEW-SESSION2  PIC X(9) VALUE
         X'455353494F4E5F434F'.
   15 MQCI-NEW-SESSION3  PIC X(6) VALUE
         X'5252454C4944'.
```

MQIs Supported by VISION:Results

The MQIs that are supported by VISION:Results are listed below and further explained in the sections that follow.

- MQCONN connects the application to a WebSphere MQ Queue manager.
- MQDISC disconnects the application from a queue manager.
- MQOPEN opens a specific queue for message access.
- MQCLOSE closes a specific queue.
- MQGET retrieves a message from a queue.
- MQPUT places a message into a queue.
- MQPUT1 serves as a combination of MQOPEN, MQPUT, and MQCLOSE to handle a single message for a specified queue.
- MQINQ inquires about the properties of an object.
- MQSET sets properties of an object.
- MQCONNX connects with standard or FastPath bindings.
- MQBEGIN begins a unit of work (coordinates with DB2 or IMS).

  Note: This MQI is not supported by IBM for the z/OS environment.
- MQCMIT commits a unit of work.
- MQBACK backs out a unit of work.

MQCONN

The purpose of MQCONN is to CONNECT to the queue manager.

Syntax

```
MQCONN qmgr hconn compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
qmgr	Data name that contains the name of the MQSeries queue manager that the application will use.
hconn	Data name that contains the connection handle returned from a successful CONNECT.
compcode	Data name that contains the completion code returned from a CONNECT request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from a CONNECT request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If errormsg is omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a U700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy book, *CMQWORK*, contains the data definitions of the data names that should be used whenever MQCONN is used.

MQDISC

The purpose of MQDISC is to DISCONNECT from the queue manager.

Syntax

```
MQDISC hconn compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle established from the successful CONNECT.
compcode	Data name that contains the completion code returned from a DISCONNECT request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from a DISCONNECT request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a U700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy book, *CMQWORK*, contains the data definitions of the data names that should be used whenever MQDISC is used.

MQOPEN

The purpose of MQOPEN is to OPEN access to the object (queue, NameList, process definition, or queue manager) that has been connected.

Syntax

```
MQOPEN  hconn mqod hoptions hobj compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
mqod	Data name that contains the description of the object to open.
hoptions	Data name that contains the options controlling the open.
hobj	Data name that contains the object handle for the opened object.
compcode	Data name that contains the completion code returned from an OPEN request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from an OPEN request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a U700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy books, *CMQWORK* and *CMQODV*, contain the data definitions of the data names that should be used whenever MQOPEN is used.

MQCLOSE

The purpose of MQCLOSE is to CLOSE (relinquish) access to the object that has been opened.

Syntax

```
MQCLOSE hconn hobj hoptions compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
hobj	Data name that contains the object handle for the object (queue) to close.
hoptions	Data name that contains the options controlling the close.
compcode	Data name that contains the completion code returned from the CLOSE request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the CLOSE request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy book, *CMQWORK*, contains the data definitions of the data names that should be used whenever MQCLOSE is used.

MQGET

The purpose of MQGET is to GET the first or next message from a particular queue, which has already been opened. See MQGET Sample Source Program on page 52 for a sample of the program source. See MQGET Sample Compiled Listing on page 67 for a sample of the compiled listing.

Syntax

```
MQGET  hconn hobj mqmd mqgmo bufferlen buffer datalen compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
hobj	Data name that contains the object handle for the object (queue or distribution list) to retrieve.
mqmd	Data name that contains the message descriptor attributes for the message being retrieved.
mqgmo	Data name that contains the options controlling the retrieval.
bufferlen	Data name that contains the total length of the message buffer.
buffer	Data name that contains the message to be retrieved.
datalen	Data name that contains the length of the application data in the message buffer.
compcode	Data name that contains the completion code returned from the GET request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the GET request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages. The errormsg field should be coded when browsing and obtaining all the messages from a queue. When no more messages are found in the queue, WebSphere MQ returns a completion code of 2 and a reason code of 2033. Because a completion code of 2 is defined as a WebSphere MQ error, VISION:Results prints an error message in the listings and ends the program with a u700. However, by coding the errormsg field as the last parameter on the MQI call, you prevent the error message from printing and the program from ending with a u700.

Note: The COBOL copy books, CMQWORK, CMQMDV, and CMQGMOV, contain the data definitions of the data names that should be used whenever MQGET is used.

MQPUT

The purpose of MQPUT is to PUT a message on a particular queue or distribution list, which has already been opened. See MQPUT Sample Source Program on page 61 for a sample of the program source. See MQPUT Sample Compiled Listing on page 101 for a sample of the compiled listing.

Syntax

```
MQPUT hconn hobj mqmd mqpmo bufferlen buffer compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data data namename that contains the connection handle returned from a successful CONNECT.
hobj	Data name that contains the object handle for the object (queue or distribution list) to update.
mqmd	Data name that contains the message descriptor attributes for the message being added.
mqpmo	Data name that contains the options controlling the PUT.
bufferlen	Data name that contains the total length of the message buffer.
buffer	Data name that contains the message to be added.
compcode	Data name that contains the completion code returned from the PUT request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the PUT request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy books, CMQWORK, CMQMDV, and CMQPMOV, contain the data definitions of the data names that should be used whenever MQPUT is used.

MQPUT1

The purpose of MQPUT1 is to PUT a message on a particular queue. MQPUT1 opens a queue, writes a single message, and then closes the queue.

Syntax

```
MQPUT1  hconn mqod mqmd mqpmo bufferlen buffer compcode reason  [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
mqod	Data name that contains the object queue descriptors for the message queue being updated.
mqmd	Data name that contains the message descriptor attributes for the message being added.
mqpmo	Data name that contains the options controlling the PUT.
bufferlen	Data name that contains the total length of the message buffer.
buffer	Data name that contains the message to be added.
compcode	Data name that contains the completion code returned from the PUT1 request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the PUT1 request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy books, CMQWORK, CMQODV, CMQMDV, and CMQPMOV, contain the data definitions of the data names that should be used whenever MQPUT1 is used.

MQINQ

The purpose of MQINQ is to INQ (inquire) about an object's attributes.

Syntax

```
MQINQ  hconn hobj  selectorcount selectorstable intattrcount intattrstable
charattrlength charattrs compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
hobj	Data name that contains the object handle for the object (queue or distribution list) to update.
selectorcount	Data name that contains the count of attribute selectors for inquiry.
selectorstable	Data name that contains the array of attribute selectors for inquiry.
intattrcount	Data name that contains the count of integer attributes for inquiry.
intattrstable	Data name that contains the array of integer attributes for the inquiry.
charattrlength	Data name that contains the length of the character attribute buffer for the inquiry.
charattrs	Data name that contains character attributes for the inquiry.
compcode	Data name that contains the completion code returned from the INQ request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from INQ request.
errormsg	Data name that contains DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does error handling, which involves displaying DYL-1207E error message in listings and ending the program with a u700 ABEND. When this parameter is present, the program determines whether to print errormsg field and return code that should be set when program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy book, CMQWORK, contains the data definitions of the data names that should be used whenever MQINQ is used.

MQSET

The purpose of MQSET is to SET the property or queue attributes for an object.

Syntax

```
MQSET  hconn hobj selectorcount selectorstable intattrcount intattrstable
charattrlength charattrs compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
hobj	Data name that contains the object handle for the object (queue or distribution list) to update.
selectorcount	Data name that contains the count of attribute selectors for this SET function.
selectorstable	Data name that contains the array of attribute selectors for this SET function.
intattrcount	Data name that contains the count of integer attributes for this SET function.
intattrstable	Data name that contains the array of integer attributes for this SET function.
charattrlength	Data name that contains the length of the character attribute buffer for this SET function.
charattrs	Data name that contains the character attributes for this SET function.
compcode	Data name that contains the completion code returned from the SET request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the SET request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy book, CMQWORK, contains the data definitions of the data Data namenames that should be used whenever MQSET is used.

MQCONNX

The purpose of CONNECTX is to CONNECT to a queue manager with the ability to specify options controlling the CALL.

Syntax

```
MQCONNX  qmgr  mqcno  hconn  compcode  reason [errormsg]
```

Data Field Definitions

Data Field	Definition
qmgr	Data name that contains the queue manager name to connect.
mqcno	Data name that contains the options to control the CONNX.
hconn	Data name that contains the connection handle returned from a successful CONNECTX.
compcode	Data name that contains the completion code returned from the CONNECTX request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the CONNECTX request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy books, CMQWORK and CMQCNOV, contain the data definitions of the data names that should be used whenever MQCONNX is used.

MQBEGIN

MQBEGIN is not a supportable MQI for either z/OS or VSE. The MQBEGIN syntax is provided here to preserve compatibility. If coded, the proper number of parameters must be specified, or a validation error will occur.

Syntax

```
MQBEGIN  hconn mqbo compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
mqbo	Data name that contains the BEGIN options.
compcode	Data name that contains the completion code returned from the BEGIN request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the BEGIN request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

MQCMIT

The purpose of MQCMIT is to COMMIT a unit of work.

Syntax

```
MQCMIT  hconn compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
compcode	Data name that contains the completion code returned from the COMMIT request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the COMMIT request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy book, CMQWORK, contains the data definitions of the data names that should be used whenever MQCMIT is used.

MQBACK

The purpose of MQBACK is to BACKOUT (a unit of work) all message GETS and PUTS since last syncpoint (usually IMS or DB2).

Syntax

```
MQBACK  hconn compcode reason [errormsg]
```

Data Field Definitions

Data Field	Definition
hconn	Data name that contains the connection handle returned from a successful CONNECT.
compcode	Data name that contains the completion code returned from the BACKOUT request. This field must be checked to determine whether to terminate or to continue the program.
reason	Data name that contains the reason code returned from the BACKOUT request.
errormsg	Data name that contains the DYL-1207E error message from the MQI. This parameter is optional. If omitted, VISION:Results does the error handling, which involves displaying the DYL-1207E error message in the listings and ending the program with a u700 ABEND. When this parameter is present, the program will determine whether to print the errormsg field and the return code that should be set when the program terminates. VISION:Results does error handling for all other messages.

Note: The COBOL copy book, CMQWORK, contains the data definitions of the data names that should be used whenever MQBACK is used.

Error Handling

If the errormsg field has been coded, the MQI's generated completion code and reason code is available for you to view after each MQI call. A brief text message of the completion code and reason code is provided in the errormsg field. For a more complete explanation of the error, see the *WebSphere MQ for z/OS Messages and Codes* manual.

If the errormsg field has not been coded, VISION:Results handles the error. The completion code and reason code along with a brief text message is printed in the listings, and the program is set to end with a u700.

The errormsg field should be coded when browsing and obtaining all the messages from a queue. When no more messages are found in the queue, WebSphere MQ returns a completion code of 2 and a reason code of 2033. Because a completion code of 2 is defined as a WebSphere MQ error, VISION:Results prints an error message in the listings and end the program with a u700. However, by coding the errormsg field as the last parameter on the MQI call, you prevent the error message from printing and the program from ending with a u700.

You can override the u700 return code, when the errormsg is not the last parameter on the MQI call, by moving a different value into the reserved word field called DYLRETURN. However, the error message will still be displayed in the listings.

Important! *VISION:Results does not terminate the program at the time of the error; instead, the program continues processing. You must code your program to check the completion code field called COMPCODE. This field determines whether to terminate or to continue the program.*

Error Messages

The WebSphere MQ Series error messages are listed here.

Number	Message	Meaning
1207E	`An error occurred in # codes and message follow: COMPLETION CODE = #,REASON CODE=# An error text message is displayed.`	The MQI ended with an error. Make the appropriate corrections suggested by the error text message.
1270E	`MQSeries not allowed for # system.`	WebSphere MQ Series is not allowed for either the VSE or CMS systems.
1271E	`Parameter missing for MQSeries.`	Data name parameters should be coded after the WebSphere MQ Series keyword.
1272E	`MQSeries data name of # is invalid.`	A data name must start with a letter and be at least 2 characters long.
1273E	`Data namedata name of # is undefined.`	A data name following the WebSphere MQ Series keyword is not defined.
1274E	`Literal of # is invalid.`	The literal specified following the WebSphere MQ Series keyword is invalid. It must be 1 to 20 characters long.

Sample Source Programs and Compiled Listings

This section provides samples of source programs and compiled listings for MQGET and MQPUT.

MQGET Sample Source Program

```
OPTION STRUCTURED2
* ************************************************************ *
*                                                              *
*  Description      : Sample program to get and print messages  *
*                     from a specified queue.                  *
*                                                              *
*  Function         : This program prints a report showing all  *
*                     the messages in a specified queue in a   *
*                     specified queue manager                  *
*                                                              *
*                     The program processes the first 80 bytes *
*                     only of each message.  It uses the BROWSE*
*                     option of the MQGET call to ensure that  *
*                     data is not lost                         *
*                                                              *
*  Return Values    : 0 - Successful completion                *
*                     4 - Parameter error, eg: wrong number    *
*                         of parameters passed                 *
*                     8 - Error in MQ call, eg: unknown object *
*                         name                                 *
*                                                              *
* ************************************************************ *
*                                                              *
*                      Program logic                           *
*                      -------------                           *
*                                                              *
*   Start  (MAIN SECTION)                                      *
*   -----                                                      *
*                                                              *
*       Open print file                                        *
*       Print first line of header (Perform PRINT_HDR1)        *
*                                                              *
*       Obtain the input data from PARM='UA=aaa,bbb':
*          - aaa is the name of the queue manager              *
*          - bbb is the name of the queue                      *
*                                                              *
*          If the name of the queue manager is missing         *
*             Build a warning message and move it into data line*
*             Print the line (Perform PRINT-LINE)              *
*             Continue (using default queue manager name)      *
*          End-if                                              *
*                                                              *
*          If the name of the queue is missing                 *
*             Build an error message and move it into data line *
*             Print the line (Perform PRINT-LINE)              *
*             Branch to Exit2                                  *
*          End-if                                              *
*                                                              *
*       Print the rest of the header (Perform PRINT_HDR2)      *
*                                                              *
*       Connect to the queue manager                           *
*       If an error occurs                                     *
*          Build an error message and move it into data line   *
*          Print the line (Perform PRINT-LINE)                 *
*          Branch to Exit2                                     *
*       End-if                                                 *
*                                                              *
*       Open the queue                                         *
*       If an error occurs                                     *
*          Build an error message and move it into data line   *
*          Print the line (Perform PRINT-LINE)                 *
*          Branch to Exit1                                     *
*       End-if                                                 *
```

Figure 20 MQGET Sample Source Program (Page 1 of 10)

```
*                                                                *
*       Get the first message (using BROWSE-FIRST option)        *
*                                                                *
*       Do while no error                                        *
*                                                                *
*          Add 1 to relative message number                      *
*          Move message into print line (maximum 80 bytes)       *
*          Print the line (Perform PRINT-LINE)                   *
*                                                                *
*          Get next message (using BROWSE-NEXT option)           *
*                                                                *
*       End-do                                                   *
*                                                                *
*       When an error occurs                                     *
*          If no more messages                                   *
*             Do nothing                                         *
*          else                                                  *
*             Build an error message and move it into data line  *
*             Print the line (Perform PRINT-LINE)                *
*          End-if                                                *
*       End-if                                                   *
*                                                                *
*       Close the queue                                          *
*       If an error occurs                                       *
*          Build an error message and move it into data line     *
*          Print the line (Perform PRINT-LINE)                   *
*       End-if                                                   *
*                                                                *
*   Exit1  (DISCONNECT)                                          *
*   -----                                                        *
*                                                                *
*       Disconnect from the queue manager                        *
*       If an error occurs                                       *
*          Build an error message and move it into data line     *
*          Print the line (Perform PRINT-LINE)                   *
*       End-if                                                   *
*                                                                *
*   Exit2  (MAIN_END)                                            *
*   -----                                                        *
*                                                                *
*       Set the return code                                      *
*                                                                *
*       Close print file                                         *
*                                                                *
*       Stop run                                                 *
*                                                                *
*   Print line  (PRINT_LINE SECTION)                             *
*   ----------                                                   *
*                                                                *
*       If number of lines printed is greater than page maximum  *
*          Print first line of header (Perform PRINT_HDR1)       *
*          Print the rest of the header (Perform PRINT_HDR2)*
*       End-if                                                   *
*                                                                *
*       Print data line                                          *
*                                                                *
*       Add 1 to count of lines printed                          *
*                                                                *
*       Return to performing section                             *
*                                                                *
*   Print first line of header  (PRINT_HDR1 SECTION)             *
*   --------------------------                                   *
*                                                                *
*       Add 1 to page number                                     *
*                                                                *
*       Print first line after jumping to top of page            *
*                                                                *
*       Set number of lines printed to 1                         *
*                                                                *
*       Return to performing section                             *
*                                                                *
*   Print rest of header  (PRINT_HDR2 SECTION)                   *
*   --------------------                                         *
*                                                                *
*       Print remaining header lines                             *
*                                                                *
```

Figure 20 MQGET Sample Source Program (Page 2 of 10)

```
*       Return to performing section                            *
*                                                               *
* ************************************************************* *
* ------------------------------------------------------------- *
FILE XFILE DUMMY
FILE SYSOUT FB 133 OUTPUT FROM SYSOUT
PRINT_REC          133    CH
CARRIAGE_CONTROL     1  1 CH
PRINT_DATA         132  2 CH
WORKAREA
  PARMLEN            2  BI VALUE  0
  LOW_VALUES        48  CH VALUE LOWVALUES
  MESSAGE_COUNT      4  PD
  DATA_LENGTH        4  PD
  COMP_CODE          4  PD
  REASON_CODE        4  PD
WORKAREA
  PARMINFO          60     VALUE NULL  ;JCL PARM Parameter
  PARMDATA           1  1              ;PARM data one byte at a time
WORKAREA
       $COBOL
      * ------------------------------------------------------------- *
      *
      *    W00 - General work fields
      *
       01  W00-MAX-LINES               PIC S9(04) BINARY  VALUE +60.
       01  W00-LINE-COUNT              PIC S9(04) BINARY  VALUE ZERO.
       01  W00-PAGE-NUMBER             PIC S9(04) BINARY  VALUE ZERO.
       01  W00-MESSAGE-COUNT           PIC S9(09) BINARY  VALUE ZERO.
       01  W00-DATE.
           05  W00-YY                  PIC X(02).
           05  W00-MM                  PIC X(02).
           05  W00-DD                  PIC X(02).
       01  W00-PRINT-DATA              PIC X(132).
      *
      *    W01 - Lines of the print report
      *
       01  W01-HEADER-1.
           05  FILLER                  PIC X(10) VALUE SPACES.
           05  W01-DATE                PIC X(10).
           05  FILLER                  PIC X(36) VALUE SPACES.
           05  FILLER                  PIC X(19) VALUE
                                                  'SAMPLE QUEUE REPORT'.
           05  FILLER                  PIC X(38) VALUE SPACES.
           05  FILLER                  PIC X(05) VALUE 'PAGE '.
           05  W01-PAGE                PIC ZZZ9.
           05  FILLER                  PIC X(10) VALUE SPACES.
       01  W01-HEADER-2.
           05  FILLER                  PIC X(25) VALUE SPACES.
           05  FILLER                  PIC X(29) VALUE
                                       '        QUEUE MANAGER NAME : '.
           05  W01-MQM-NAME            PIC X(48) VALUE SPACES.
           05  FILLER                  PIC X(30) VALUE SPACES.
       01  W01-HEADER-3.
           05  FILLER                  PIC X(37) VALUE SPACES.
           05  FILLER                  PIC X(17) VALUE
                                                    '    QUEUE NAME : '.
           05  W01-QUEUE-NAME          PIC X(48) VALUE SPACES.
           05  FILLER                  PIC X(30) VALUE SPACES.
       01  W01-HEADER-4.
           05  FILLER                  PIC X(16)  VALUE SPACES.
           05  FILLER                  PIC X(116) VALUE '  RELATIVE'.
       01  W01-HEADER-5.
           05  FILLER                  PIC X(16)  VALUE SPACES.
           05  FILLER                  PIC X(10)  VALUE '   MESSAGE'.
           05  FILLER                  PIC X(106) VALUE '   MESSAGE'.
       01  W01-HEADER-6.
           05  FILLER                  PIC X(16) VALUE SPACES.
           05  FILLER                  PIC X(10) VALUE '   NUMBER '.
           05  FILLER                  PIC X(10) VALUE '   LENGTH '.
           05  FILLER                  PIC X(96) VALUE
                          '--------------------------------- MESSAGE DATA
      -                   ' ---------------------------------'.
       01  W01-REPORT-LINE.
           05  FILLER                  PIC X(16) VALUE SPACES.
           05  W01-MESSAGE-NUMBER      PIC Z(8)9.
```

Figure 20 MQGET Sample Source Program (Page 3 of 10)

```
           05  FILLER                  PIC X     VALUE SPACE.
           05  W01-MESSAGE-LENGTH      PIC Z(8)9.
           05  FILLER                  PIC X     VALUE SPACE.
           05  W01-DATA                PIC X(80).
           05  FILLER                  PIC X(16) VALUE SPACES.
      *
      *    W02 - Data fields derived from the PARM field
      *
       01  W02-MQM                     PIC X(48) VALUE SPACES.
       01  W02-OBJECT                  PIC X(48) VALUE SPACES.
      *
      *    W04 - Error and information messages
      *
       01  W04-MESSAGE-0.
           05  FILLER              PIC X(48) VALUE SPACES.
           05  FILLER              PIC X(35) VALUE
                                  '********** END OF REPORT **********'.
           05  FILLER              PIC X(49) VALUE SPACES.
       01  W04-MESSAGE-1.
           05  FILLER              PIC X(10)  VALUE SPACES.
           05  FILLER              PIC X(122) VALUE
              '********** NO DATA PASSED TO PROGRAM. PROGRAM REQUIRES A
      -       'QUEUE MANAGER NAME AND A QUEUE NAME. **********'.
       01  W04-MESSAGE-2.
           05  FILLER              PIC X(25)  VALUE SPACES.
           05  FILLER              PIC X(107) VALUE
              '********** NO QUEUE MANAGER NAME PASSED TO PROGRAM - DEFA
      -       'ULT USED *****'.
       01  W04-MESSAGE-3.
           05  FILLER              PIC X(38) VALUE SPACES.
           05  FILLER              PIC X(94) VALUE
              '********** NO QUEUE NAME PASSED TO PROGRAM. **********'.
       01  W04-MESSAGE-4.
           05  FILLER              PIC X(13) VALUE SPACES.
           05  FILLER              PIC X(32) VALUE
                   '********** AN ERROR OCCURRED IN '.
           05  W04-MSG4-TYPE       PIC X(10).
           05  FILLER              PIC X(20) VALUE
                   '. COMPLETION CODE = '.
           05  W04-MSG4-COMPCODE   PIC Z(8)9.
           05  FILLER              PIC X(15) VALUE ' REASON CODE ='.
           05  W04-MSG4-REASON     PIC Z(8)9.
           05  FILLER              PIC X(24) VALUE ' **********'.
      *
      *    W06 - Return values
      *
       01  W06-CSQ4-OK             PIC S9(4) VALUE 0.
       01  W06-CSQ4-WARNING        PIC S9(4) VALUE 4.
       01  W06-CSQ4-ERROR          PIC S9(4) VALUE 8.
       $ECOBOL
*
*    The following copy files define API control blocks.
*
WORKAREA
COPY CMQWORK COBOL
WORKAREA
COPY CMQCNOV COBOL
WORKAREA
COPY CMQMDV COBOL
WORKAREA
COPY CMQGMOV COBOL
WORKAREA
COPY CMQODV COBOL
WORKAREA
COPY CMQPMOV COBOL
MAIN:
* ---------------------------------------------------------------- *
*                                                                  *
* This section receives the names of the queue manager and the     *
* queue from the PARM statement in the JCL. It opens the queue,    *
* reads all the messages, and prints them                          *
*                                                                  *
* This section uses the MQGET call with the BROWSE option to       *
* ensure that the data is not removed from the queue               *
*                                                                  *
* ---------------------------------------------------------------- *
```

Figure 20 MQGET Sample Source Program (Page 4 of 10)

```
*
*    Open the print file, initialize the fields for the
*    header date and the page number, and print the first
*    line of the header
*
*
     MOVE DYLDATE4 TO  W01_DATE
*
     PERFORM PRINT_HDR1
*
*
*    Retrieved Queue Manager Name and Queue Name from JCL PARM statement
*
*    Following fields are initialized
*      PARMINFO =  Data retrieved from PARM statement
*      INX      =  Index used to find delimiter
*      INZ      =  Index used to calculate size of PARM value
*      INY      =  Index used to determine number of PARMS retrieved
*
*
     MOVE DYLPARM    TO PARMINFO
     MOVE 0          TO INX
     MOVE 0          TO INZ
     MOVE 1          TO INY
*
*    Separate into the relevant fields any data passed in the
*    PARM statement. The fields are separated by a comma. The last field
*    ends with a blank.
*
     DOWHILE INX LE 60 AND INY LE 2
       IF PARMDATA(INX) EQ ',' OR PARMDATA(INX) EQ ' '
         PARMLEN = INX - INZ              ;Length of PARM data value
         CASE INY                         ;Determine current PARM field
           WHEN EQ 1                      ;Queue Manager Name?
            MOVE PARMDATA(INZ)
             TO W02_MQM LENGTH PARMLEN    ;Save Queue Manager Name
           WHEN EQ 2                      ;Queue Name?
            MOVE PARMDATA(INZ)
             TO W02_OBJECT LENGTH PARMLEN ;Save Queue Name
         ENDCASE
         INY = INY + 1                    ;Next PARM data field number
       ENDIF
       IF PARMDATA(INX) NE ' '            ;Not the last PARM data field
         IF PARMDATA(INX) EQ ','          ;End of current PARM data
           INZ = INX + 1                  ;Start of new PARM data field
         ENDIF
         INX = INX + 1                    ;Check next parameter byte
       ENDIF
     ENDDO
*
*    If no data was passed, create a message, print it, and
*    exit
*
      IF INX GT 60
        MOVE W04_MESSAGE_1 TO W00_PRINT_DATA
        PERFORM PRINT_LINE
        MOVE W06_CSQ4_WARNING TO DYLRETURN
        GOTO MAIN_END
      ENDIF
*
*    Move the data (spaces if nothing is entered) into the
*    relevant API parameter fields
*
      MOVE W02_MQM    TO QMGR
*
*    Move the data (spaces if nothing is entered) into the
*    relevant print fields
*
      MOVE W02_MQM    TO W01_MQM_NAME
      MOVE W02_OBJECT TO W01_QUEUE_NAME
*
*    Print a message if the queue manager name is missing, the
*    default queue manager will be used
*
     IF W02_MQM EQ SPACES OR W02_MQM EQ LOW_VALUES
        MOVE W04_MESSAGE_2 TO W00_PRINT_DATA
```

Figure 20 MQGET Sample Source Program (Page 5 of 10)

```
           PERFORM PRINT_LINE
        ENDIF
*
*       Print a message if the queue name is missing and exit from
*       program
*
        IF W02_OBJECT EQ SPACES OR W02_OBJECT EQ LOW_VALUES
           MOVE W04_MESSAGE_3 TO W00_PRINT_DATA
           PERFORM PRINT_LINE
           MOVE W06_CSQ4_WARNING TO DYLRETURN
           GOTO MAIN_END
        ENDIF
*
*       Print the remaining header lines
*
        PERFORM PRINT_HDR2
*
*       Connect to the specified queue manager.
*
        MQCONN    QMGR                  ;Queue Manager Name
                  HCONN                 ;Connection Handle
                  COMPCODE              ;Completion Code
                  REASON                ;Reason Code
*
*       Test the output of the connect call.  If the call failed,
*       print an error message showing the completion code and
*       reason code
*
        IF COMPCODE NE MQCC_OK
           MOVE 'CONNECT'     TO W04_MSG4_TYPE
           MOVE COMPCODE      TO COMP_CODE
           MOVE REASON        TO REASON_CODE
           EDIT COMP_CODE   INTO W04_MSG4_COMPCODE USING Z
           EDIT REASON_CODE INTO W04_MSG4_REASON USING Z
           MOVE W04_MESSAGE_4 TO W00_PRINT_DATA
           PERFORM PRINT_LINE
           MOVE W06_CSQ4_ERROR TO DYLRETURN
           GOTO MAIN_END
        ENDIF
*
*       Initialize the object descriptor (MQOD) control block.
*       (The copy file initializes all the other fields)
*
        MOVE MQOT_Q            TO MQOD_OBJECTTYPE
        MOVE W02_OBJECT        TO MQOD_OBJECTNAME
*
*       Initialize the working storage fields required to open
*       the queue
*
*         HOPTIONS is set to open the queue for browsing
*         HOBJ     is set by the MQOPEN call and is used by the
*                  MQGET and MQCLOSE calls
*
        MOVE MQOO_BROWSE       TO HOPTIONS
*
*       Open the queue.
*
        MQOPEN    HCONN                 ;Connection Handle
                  MQOD                  ;Object Description
                  HOPTIONS              ;Control Options
                  HOBJ                  ;Object Handle
                  COMPCODE              ;Completion Code
                  REASON                ;Reason Code
*
*       Test the output of the open call.  If the call failed, print
*       an error message showing the completion code and reason code
*
        IF COMPCODE NE MQCC_OK
           MOVE 'OPEN'        TO W04_MSG4_TYPE
           MOVE COMPCODE      TO W04_MSG4_COMPCODE
           MOVE REASON        TO W04_MSG4_REASON
           MOVE W04_MESSAGE_4 TO W00_PRINT_DATA
           PERFORM PRINT_LINE
           MOVE W06_CSQ4_ERROR TO DYLRETURN
           GOTO DISCONNECT
        ENDIF
```

Figure 20 MQGET Sample Source Program (Page 6 of 10)

```
*
*    No need to change the Message Descriptor (MQMD) control
*    block because the copy file initializes all the fields
*
*    Initialize the Get Message Options (MQGMO) control block.
*    (The copy file initializes all the other fields)
*
     MQGMO_OPTIONS = MQGMO_NO_WAIT + MQGMO_ACCEPT_TRUNCATED_MSG
     MQGMO_OPTIONS = MQGMO_OPTIONS + MQGMO_BROWSE_FIRST
*
*    Make the first get call outside the loop because this call
*    uses the BROWSE-FIRST option
*
     MQGET     HCONN                  ;Connection Handle
               HOBJ                   ;Object Handle
               MQMD                   ;Messae Descriptor Attributes
               MQGMO                  ;Retrieval Options
               BUFFERLEN              ;Message Buffer Length
               BUFFER                 ;Message Buffer
               DATALEN                ;Message Buffer's Data Length
               COMPCODE               ;Completion Code
               REASON                 ;Reason Code
*
*    Test the output of the get call using the PERFORM loop
*    that follows.
*
*    Change the MQGMO Options field to BROWSE-NEXT.
*
     MQGMO_OPTIONS = MQGMO_NO_WAIT + MQGMO_ACCEPT_TRUNCATED_MSG
     MQGMO_OPTIONS = MQGMO_OPTIONS + MQGMO_BROWSE_NEXT
*
*    Loop until the get call fails
*    - we test for call not successful and the one condition
*      after which we want to continue within the loop
*      (the received message has been truncated)
*
     DOWHILE COMPCODE EQ MQCC_OK OR
            (COMPCODE EQ MQCC_WARNING AND
             REASON EQ MQRC_TRUNCATED_MSG_ACCEPTED)
*
*       Increment the relative message number.  Move the message
*       number and the first 80 bytes of the message data into
*       the print line
*
        W00_MESSAGE_COUNT = W00_MESSAGE_COUNT + 1
        MOVE W00_MESSAGE_COUNT TO MESSAGE_COUNT
        MOVE DATALEN           TO DATA_LENGTH
        EDIT MESSAGE_COUNT   INTO W01_MESSAGE_NUMBER USING Z
        EDIT DATA_LENGTH     INTO W01_MESSAGE_LENGTH USING Z
        MOVE BUFFER            TO W01_DATA
        MOVE W01_REPORT_LINE   TO W00_PRINT_DATA
*
*       Print the message line
*
        PERFORM PRINT_LINE
*
*****************************************************************
*    MQMD-MSGID and MQMD-CORRELID are input/output fields that  *
*    are filled and read by MQGET.  Clear them before the next  *
*    MQGET call to ensure that all messages are retrieved.      *
*****************************************************************
*
        MOVE MQMI_NONE TO MQMD_MSGID
        MOVE MQCI_NONE TO MQMD_CORRELID
*
*    Clear the message data field before the next get call to
*    ensure that no old data remains if the next line is shorter
*
        MOVE SPACES TO BUFFER
*
*       Get the next message
*
        MQGET     HCONN               ;Connection Handle
                  HOBJ                ;Object Handle
                  MQMD                ;Messae Descriptor Attributes
                  MQGMO               ;Retrieval Options
```

Figure 20 MQGET Sample Source Program (Page 7 of 10)

```
                  BUFFERLEN           ;Message Buffer Length
                  BUFFER              ;Message Buffer
                  DATALEN             ;Message Buffer's Data Length
                  COMPCODE            ;Completion Code
                  REASON              ;Reason Code
                  ERRORMSG            ;Error Message
*
*       Test the output of the MQGET call at the top of the loop.
*       Exit the loop if an error occurs
*
           ENDDO
*
*    Test the output of the get call
*
*    When the loop reaches the end of the messages, the
*    completion code is MQCC-FAILED and the reason code
*    is MQRC-NO-MSG-AVAILABLE
*
*    If the call failed for any other reason,
*    print an error message showing the completion code and
*    reason code
*
     IF COMPCODE EQ MQCC_FAILED AND
        REASON EQ MQRC_NO_MSG_AVAILABLE
*
        MOVE W04_MESSAGE_0 TO W00_PRINT_DATA
*
     ELSE
        MOVE 'GET'         TO W04_MSG4_TYPE
        MOVE COMPCODE      TO W04_MSG4_COMPCODE
        MOVE REASON        TO W04_MSG4_REASON
        MOVE W04_MESSAGE_4 TO W00_PRINT_DATA
     ENDIF
*
     PERFORM PRINT_LINE
*
* Close the queue
*
     MOVE MQCO_NONE TO HOPTIONS
*
     MQCLOSE      HCONN               ;Connection Handle
                  HOBJ                ;Object Handle
                  HOPTIONS            ;Control Options
                  COMPCODE            ;Completion Code
                  REASON              ;Reason Code
*
*    Test the output of the MQCLOSE call.  If the call failed,
*    print an error message showing the completion code and reason
*    code
*
     IF COMPCODE NE MQCC_OK
        MOVE 'CLOSE'       TO W04_MSG4_TYPE
        MOVE COMPCODE      TO W04_MSG4_COMPCODE
        MOVE REASON        TO W04_MSG4_REASON
        MOVE W04_MESSAGE_4 TO W00_PRINT_DATA
        PERFORM PRINT_LINE
        MOVE W06_CSQ4_ERROR TO DYLRETURN
     ENDIF
*
DISCONNECT:
*
* Disconnect from the queue manager
*
     MQDISC    HCONN                 ;Connection Handle
               COMPCODE              ;Completion Code
               REASON                ;Reason Code
*
*    Test the output of the disconnect call.  If the call failed,
*    print an error message showing the completion code and
*    reason code
*
     IF COMPCODE NE MQCC_OK
        MOVE 'DISCONNECT'  TO W04_MSG4_TYPE
        MOVE COMPCODE      TO W04_MSG4_COMPCODE
        MOVE REASON        TO W04_MSG4_REASON
        MOVE W04_MESSAGE_4 TO W00_PRINT_DATA
```

Figure 20 MQGET Sample Source Program (Page 8 of 10)

```
        MOVE W06_CSQ4_ERROR TO DYLRETURN
        PERFORM PRINT_LINE
     ENDIF
*
MAIN_END:
*
*    Stop the program
*
     STOP
*
*****************************************************************
*
PRINT_LINE:
*
*    This section prints all data lines produced by the program
*
*    If the maximum number of lines for a page has been printed,
*    start a new page
*
     IF W00_LINE_COUNT GT W00_MAX_LINES
        PERFORM PRINT_HDR1
        PERFORM PRINT_HDR2
     ENDIF
*
     MOVE W00_PRINT_DATA TO PRINT_DATA
     MOVE SPACE TO CARRIAGE_CONTROL
     WRITE SYSOUT
*
     W00_LINE_COUNT = W00_LINE_COUNT + 1
*
*
*****************************************************************
*
PRINT_HDR1:
*
*    This section prints the first line of the report.
*    This is separate from the section that prints the other
*    header lines because the first line is needed every time
*    the program runs
*
     W00_PAGE_NUMBER = W00_PAGE_NUMBER + 1
     MOVE W00_PAGE_NUMBER TO W01_PAGE
     MOVE W01_HEADER_1 TO PRINT_DATA
     MOVE '1'  TO  CARRIAGE_CONTROL
     WRITE SYSOUT
*
     MOVE 1 TO W00_LINE_COUNT
*
*
*****************************************************************
*
PRINT_HDR2:
*
*    This section prints the remaining header lines
*
     MOVE W01_HEADER_2 TO PRINT_DATA
     MOVE '0' TO CARRIAGE_CONTROL
     WRITE SYSOUT
     W00_LINE_COUNT = W00_LINE_COUNT + 2
*
     MOVE W01_HEADER_3 TO PRINT_DATA
     MOVE SPACE TO CARRIAGE_CONTROL
     WRITE SYSOUT
     W00_LINE_COUNT = W00_LINE_COUNT + 1
*
     MOVE W01_HEADER_4 TO PRINT_DATA
     MOVE '0' TO CARRIAGE_CONTROL
     WRITE SYSOUT
     W00_LINE_COUNT = W00_LINE_COUNT + 2
*
     MOVE W01_HEADER_5 TO PRINT_DATA
     MOVE SPACE TO CARRIAGE_CONTROL
     WRITE SYSOUT
     W00_LINE_COUNT = W00_LINE_COUNT + 1
*
     MOVE W01_HEADER_6 TO PRINT_DATA
```

Figure 20 MQGET Sample Source Program (Page 9 of 10)

```
     MOVE SPACE TO CARRIAGE_CONTROL
     WRITE SYSOUT
     W00_LINE_COUNT = W00_LINE_COUNT + 1
*
     MOVE SPACES       TO PRINT_DATA
     MOVE SPACE TO CARRIAGE_CONTROL
     WRITE SYSOUT
     W00_LINE_COUNT = W00_LINE_COUNT + 1
* ---------------------------------------------------------------
*                 End of program
* ---------------------------------------------------------------
```

Figure 20 MQGET Sample Source Program (Page 10 of 10)

MQPUT Sample Source Program

```
OPTION STRUCTURED2
* ********************************************************** *
*                                                            *
*  Description      : Sample program to put a number of      *
*                     messages to a queue.                   *
*                                                            *
*  Function         : This program writes a specified number of *
*                     messages to a specified queue in a     *
*                     specified queue manager. The messages  *
*                     would consist of a one byte character  *
*                     that would be permeated through the    *
*                     entire message.                        *
*                                                            *
*                     The queue manager, the queue name, the *
*                     number of messages, the length of the  *
*                     messages, the persistence of the       *
*                     messages, and the one byte pad character *
*                     that would be used for the message would *
*                     be passed as input via the JCL PARM    *
*                     parameter.                             *
*                                                            *
*                     Note: Since, this is a test program    *
*                           the persistence of the messages  *
*                           should be set to N.              *
*                                                            *
*  Limitation       : Maximum message length set at 32767.   *
*                                                            *
* ********************************************************** *
*                                                            *
*                       Program  Logic                       *
*                       --------------                       *
*                                                            *
*                                                            *
*  main                                                      *
*  ----                                                      *
*                                                            *
*       Obtain the input data from                           *
*        PARM='UA=aaa,bbb,ccc,d,eee,f':                      *
*          - aaa is the name of the queue manager            *
*          - bbb is the name of the queue                    *
*          - ccc is the number of messages                   *
*          - d   is the message pad character                *
*          - eee is the length of message(s)                 *
*          - f   is the persistence of message(s)            *
*                                                            *
*   Move parameters into corresponding variables.            *
*   If parameters are invalid then                           *
*            Call USAGE_ERR and exit.                        *
*                                                            *
*   Display the parameters passed to the program.            *
*                                                            *
*   Connect to the queue manager.                            *
*   If connection failed then                                *
*            Call PRINT_MSG and exit                         *
*                                                            *
```

Figure 21 MQPUT Sample Source Program (Page 1 of 6)

```
*   Open the specified message queue (MQOPEN).                  *
*   If open failed then                                         *
*            Disconnect from queue manager                      *
*            Call PRINT_MSG and exit                            *
*   Endif.                                                      *
*                                                               *
*   Set the put message options.                                *
*   Loop while the messages are put to queue                    *
*            Put message to queue (MQPUT)                       *
*            If put failed                                      *
*                     Call PRINT_MSG                            *
*                     Break from loop                           *
*            Endif                                              *
*   Endloop.                                                    *
*   Display number of messages put to the queue.                *
*                                                               *
*   Close the message queue.                                    *
*   If close failed then                                        *
*            Call PRINT_MSG.                                    *
*                                                               *
*   Disconnect from the queue manager.                          *
*   If disconnect failed then                                   *
*            Call PRINT_MSG.                                    *
*                                                               *
*   Exit program.                                               *
*                                                               *
*                                                               *
*---------------------------------------------------------------*
*                                                               *
*  USAGE_ERR                                                    *
*  -----------                                                  *
*                                                               *
*   Print message showing correct usage for program.            *
*                                                               *
*                                                               *
*---------------------------------------------------------------*
*                                                               *
*  PRINT_MSG                                                    *
*  ----------------------                                       *
*                                                               *
*   Create error message and display.                           *
*                                                               *
*                                                               *
* ************************************************************* *
* ------------------------------------------------------------- *
FILE XFILE DUMMY
WORKAREA
  PARMLEN            2  BI VALUE  0
  LOW_VALUES        48  CH VALUE LOWVALUES
  COMP_CODE          4  CH
  REASON_CODE        4  CH
  NUMBER_PUTS        4  CH
WORKAREA
  PARMINFO          60     VALUE NULL  ;JCL PARM parameter
  PARMDATA           1  1              ;PARM data one byte at a time
WORKAREA
       $COBOL
      * ------------------------------------------------------------- *
      *
      *    W00 - General work fields
      *
       01  W00-RETURN-CODE             PIC S9(4) BINARY VALUE ZERO.
       01  W00-NUMPUTS                 PIC S9(9) BINARY VALUE 0.
       01  W00-ERROR-MESSAGE           PIC X(48) VALUE SPACES.
      *
      *    Data fields derived from the PARM field
      *
       01  W00-QMGR                    PIC X(48).
       01  W00-QNAME                   PIC X(48).
       01  W00-PADCHAR                 PIC X(1) VALUE '*'.
       01  W00-MSGBUFFER.
         02  W00-MSGBUFFER-DATA        PIC X(32767).
         02  W00-MSGBUFFER-ARRAY REDEFINES W00-MSGBUFFER-DATA
                                       PIC X(1) OCCURS 32767 TIMES.
       01  W00-NUMMSGS-NUM             PIC 9(5) VALUE  0.
       01  W00-NUMMSGS-NUM-CHAR REDEFINES W00-NUMMSGS-NUM
```

Figure 21 MQPUT Sample Source Program (Page 2 of 6)

```
                                        PIC X(5).
       01  W00-NUMMSGS                  PIC S9(9) BINARY VALUE 1.
       01  W00-MSGLENGTH-NUM            PIC 9(5) VALUE 0.
       01  W00-MSGLENGTH-NUM-CHAR REDEFINES W00-MSGLENGTH-NUM
                                        PIC X(5).
       01  W00-PERSISTENCE              PIC X(1) VALUE 'N'.
           88 PERSISTENT     VALUE 'P'.
           88 NOT-PERSISTENT VALUE 'N'.
       $ECOBOL
*
*    The following copy files define API control blocks.
*
WORKAREA
COPY CMQWORK COBOL
WORKAREA
COPY CMQCNOV COBOL
WORKAREA
COPY CMQMDV COBOL
WORKAREA
COPY CMQGMOV COBOL
WORKAREA
COPY CMQODV COBOL
WORKAREA
COPY CMQPMOV COBOL
MAIN:
* ------------------------------------------------------------ *
*                                                              *
*
*    Retrieved the values for the following fields in the JCL PARM
*    statement:
*      Queue Manager Name
*      Queue Name
*      Number of Messages
*      Message Pad Character
*      Length of Message(s)
*      Persistence of Message(s)
*
*    If no parameters passed to program then
*    call USAGE_ERR and exit
*
     IF DYLPARMLEN EQ 0
        PERFORM USAGE_ERR
        MOVE 8 TO W00_RETURN_CODE
        GOTO MAIN_END
     ENDIF
*
*    Following fields are initialized
*      PARMINFO =  Data retrieved from PARM statement
*      INX      =  Index used to find delimiter
*      INZ      =  Index used to calculate size of PARM value
*      INY      =  Index used to determine number of PARMS retrieved
*
*
     MOVE DYLPARM    TO PARMINFO
     MOVE 0          TO INX
     MOVE 0          TO INZ
     MOVE 1          TO INY
*
*    Separate into the relevant fields any data passed in the
*    PARM statement. The fields are separated by a comma. The last field
*    ends with a blank.
*
     DOWHILE INX LE 60 AND INY LE 6
       IF PARMDATA(INX) EQ ',' OR PARMDATA(INX) EQ ' '
         PARMLEN = INX - INZ              ;Length of PARM data value
         CASE INY                         ;Determine current PARM field
           WHEN EQ 1                      ;Queue Manager Name?
            MOVE PARMDATA(INZ)            ;Save Queue Manager Name
             TO W00_QMGR LENGTH PARMLEN
           WHEN EQ 2                      ;Queue Name?
            MOVE PARMDATA(INZ)            ;Save Queue Name
             TO W00_QNAME LENGTH PARMLEN
           WHEN EQ 3                      ;Number of Messages?
            INW = 5 - PARMLEN
            MOVE PARMDATA(INZ)            ;Save Number of Messages
             TO W00_NUMMSGS_NUM_CHAR(INW) LENGTH PARMLEN
```

Figure 21 MQPUT Sample Source Program (Page 3 of 6)

```
            WHEN EQ 4                       ;Message Pad Character?
             MOVE PARMDATA(INZ)             ;Save Message Pad Character
              TO W00_PADCHAR LENGTH PARMLEN
            WHEN EQ 5                       ;Length of Message(s)?
             INW = 5 - PARMLEN
             MOVE PARMDATA(INZ)             ;Save Length of Message(s)
              TO W00_MSGLENGTH_NUM_CHAR(INW) LENGTH PARMLEN
            WHEN EQ 6                       ;Persistence of Message(s)?
             MOVE PARMDATA(INZ)             ;Save Persistence of Messages
              TO W00_PERSISTENCE LENGTH PARMLEN
          ENDCASE
          INY = INY + 1                     ;Next PARM data field number
        ENDIF
        IF PARMDATA(INX) NE ' '             ;Not the last PARM data field
          IF PARMDATA(INX) EQ ','           ;End of current PARM data
            INZ = INX + 1                   ;Start of new PARM data field
          ENDIF
          INX = INX + 1                     ;Check next parameter byte
        ENDIF
      ENDDO
*
*     Move the data (spaces if nothing is entered) into the
*     relevant API parameter fields
*
       MOVE W00_QMGR          TO QMGR
       MOVE W00_MSGLENGTH_NUM TO BUFFERLEN
       MOVE W00_NUMMSGS_NUM   TO W00_NUMMSGS
*
*     Display parameters to be used in the program
*
      PRINT   '=========================================='
      PRINT   'PARAMETERS PASSED :'
      PRINT   '   QMGR        - ', QMGR
      PRINT   '   QNAME       - ', W00_QNAME
      PRINT   '   NUMMSGS     - ', W00_NUMMSGS_NUM
      PRINT   '   PADCHAR     - ', W00_PADCHAR
      PRINT   '   MSGLENGTH   - ', W00_MSGLENGTH_NUM
      PRINT   '   PERSISTENCE - ', W00_PERSISTENCE
      PRINT   '=========================================='
*
*     Setup the message buffer
*
      MOVE 0          TO INX
      MOVE BUFFERLEN  TO INY
*
      DOWHILE INX LE INY
*
          MOVE W00_PADCHAR TO BUFFER(INX)
          INX = INX + 1
*
      ENDDO
*
*     Connect to the specified queue manager.
*
      MQCONN    QMGR                  ;Queue Manager Name
                HCONN                 ;Connection Handle
                COMPCODE              ;Completion Code
                REASON                ;Reason Code
*
*     If connection failed then display error message
*     and exit
*
      IF COMPCODE NE MQCC_OK
         MOVE 'MQCONN'   TO W00_ERROR_MESSAGE
         PERFORM PRINT_MSG
         MOVE REASON     TO W00_RETURN_CODE
         GOTO MAIN_END
      ENDIF
      PRINT  'MQCONN SUCCESSFUL'
*
*
*     Open the queue for output
*
      MOVE MQOO_OUTPUT TO HOPTIONS
      MOVE W00_QNAME   TO MQOD_OBJECTNAME
*
```

Figure 21 MQPUT Sample Source Program (Page 4 of 6)

```
     MQOPEN    HCONN                  ;Connection Handle
               MQOD                   ;Object Description
               HOPTIONS               ;Control Options
               HOBJ                   ;Object Handle
               COMPCODE               ;Completion Code
               REASON                 ;Reason Code
*
*    If open failed then display error message
*    and exit.
*
     IF COMPCODE NE MQCC_OK
        MOVE 'MQOPEN'   TO W00_ERROR_MESSAGE
        MOVE COMPCODE   TO COMP_CODE
        MOVE REASON     TO REASON_CODE
        PERFORM PRINT_MSG
        MOVE REASON     TO W00_RETURN_CODE
        GOTO DISCONNECT
     ENDIF
     PRINT  'MQOPEN SUCCESSFUL'
*
*
*    Set persistence depending on parameter passed
*
     IF W00_PERSISTENCE EQ 'P'
        MOVE MQPER_PERSISTENT     TO MQMD_PERSISTENCE
     ELSE
        MOVE MQPER_NOT_PERSISTENT TO MQMD_PERSISTENCE
     ENDIF
*
*    Loop until specified number of messages put to queue
*
     MOVE 0            TO INX
     MOVE W00_NUMMSGS  TO INY
*
     DOWHILE INX LT INY
*
         MOVE MQMI_NONE TO MQMD_MSGID
         MOVE MQCI_NONE TO MQMD_CORRELID
*
         MQPUT     HCONN               ;Connection Handle
                   HOBJ                ;Object Handle
                   MQMD                ;Messae Descriptor Attributes
                   MQPMO               ;Put Options
                   BUFFERLEN           ;Message Buffer Length
                   BUFFER              ;Message Buffer
                   COMPCODE            ;Completion Code
                   REASON              ;Reason Code
*
*        If put failed then display error message
*        and break out of loop
*
         IF COMPCODE NE MQCC_OK
            MOVE 'MQPUT'     TO W00_ERROR_MESSAGE
            PERFORM PRINT_MSG
            MOVE W00_NUMMSGS TO INX
            MOVE REASON      TO W00_RETURN_CODE
         ELSE
            PRINT  'MQPUT SUCCESSFUL'
            W00_NUMPUTS = W00_NUMPUTS + 1
            INX         = INX + 1
         ENDIF
*
     ENDDO
*
*    Display the number of messages successfully put
*    to the queue
*
     MOVE W00_NUMPUTS TO NUMBER_PUTS
     PRINT NUMBER_PUTS, ' MESSAGES PUT TO QUEUE'
*
* Close the queue
*
     MOVE MQCO_NONE TO HOPTIONS
*
     MQCLOSE      HCONN               ;Connection Handle
                  HOBJ                ;Object Handle
```

Figure 21 MQPUT Sample Source Program (Page 5 of 6)

```
                HOPTIONS            ;Control Options
                COMPCODE            ;Completion Code
                REASON              ;Reason Code
*
*    Test the output of the MQCLOSE call.  If the call failed,
*    print an error message
*
     IF COMPCODE NE MQCC_OK
        MOVE 'MQCLOSE'  TO W00_ERROR_MESSAGE
        PERFORM PRINT_MSG
        MOVE REASON TO W00_RETURN_CODE
     ELSE
        PRINT 'MQCLOSE SUCCESSFUL'
     ENDIF
*
DISCONNECT:
*
* Disconnect from the queue manager
*
     MQDISC    HCONN                 ;Connection Handle
               COMPCODE              ;Completion Code
               REASON                ;Reason Code
*
*    Test the output of the disconnect call.  If the call failed,
*    print an error message
*
     IF COMPCODE NE MQCC_OK
        MOVE 'MQDISC'   TO W00_ERROR_MESSAGE
        PERFORM PRINT_MSG
        MOVE REASON     TO W00_RETURN_CODE
     ELSE
        PRINT 'MQDISC SUCCESSFUL'
     ENDIF
*
MAIN_END:
*
*    Set the return code
*
     MOVE W00_RETURN_CODE TO DYLRETURN
*
     STOP
*
* ------------------------------------------------------------ *
USAGE_ERR:
* ------------------------------------------------------------ *
*
     PRINT '================================================='
     PRINT 'PARAMETERS FOR PROGRAM :'
     PRINT '     QMGR        - QUEUE MANAGER'
     PRINT '     QNAME       - QUEUE NAME'
     PRINT '     NUMMSGS     - NUMBER OF MESSAGES'
     PRINT '     PADCHAR     - MESSAGE PADDING CHARACTER'
     PRINT '     MSGLENGTH   - LENGTH OF MESSAGE(S)'
     PRINT '     PERSISTENCE - PERSISTENCE OF MESSAGE(S)'
     PRINT '================================================='
*
*
* ------------------------------------------------------------ *
PRINT_MSG:
* ------------------------------------------------------------ *
*
     MOVE COMPCODE   TO COMP_CODE
     MOVE REASON     TO REASON_CODE
     PRINT '*********************************************'
     PRINT '* ', W00_ERROR_MESSAGE
     PRINT '* COMPLETION CODE : ', COMP_CODE
     PRINT '* REASON CODE     : ', REASON_CODE
     PRINT '*********************************************'
*
* ------------------------------------------------------------ *
*                       END OF PROGRAM
* ------------------------------------------------------------ *
```

Figure 21 MQPUT Sample Source Program (Page 6 of 6)

MQGET Sample Compiled Listing

As shown in the compiled listing (Figure 22), the MQGET program is executed using the following JCL PARM parameter as input:

```
PARM='UA=CSQ1,VISION.RESULTS.TEST.QUEUE'
```

```
COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE     1
         1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
         OPTION STRUCTURED2                                                                      1
         * ************************************************************ *                      2
         *                                                              *                      3
         *  Description      : Sample program to get and print messages  *                      4
         *                     from a specified queue.                  *                      5
         *                                                              *                      6
         *  Function         : This program prints a report showing all  *                      7
         *                     the messages in a specified queue in a   *                      8
         *                     specified queue manager                  *                      9
         *                                                              *                     10
         *                     The program processes the first 80 bytes *                     11
         *                     only of each message.  It uses the BROWSE *                    12
         *                     option of the MQGET call to ensure that  *                     13
         *                     data is not lost                         *                     14
         *                                                              *                     15
         *  Return Values    : 0 - Successful completion                *                     16
         *                     4 - Parameter error, eg: wrong number    *                     17
         *                         of parameters passed                 *                     18
         *                     8 - Error in MQ call, eg: unknown object *                     19
         *                         name                                 *                     20
         *                                                              *                     21
         * ************************************************************ *                     22
         *                                                              *                     23
         *                    Program logic                             *                     24
         *                    -------------                             *                     25
         *                                                              *                     26
         *   Start  (MAIN SECTION)                                      *                     27
         *   -----                                                      *                     28
         *                                                              *                     29
         *      Open print file                                         *                     30
         *      Print first line of header (Perform PRINT_HDR1)         *                     31
         *                                                              *                     32
         *      Obtain the input data from PARM='UA=aaa,bbb':                                 33
         *         - aaa is the name of the queue manager               *                     34
         *         - bbb is the name of the queue                       *                     35
         *                                                              *                     36
         *         If the name of the queue manager is missing          *                     37
         *            Build a warning message and move it into data line*                     38
         *            Print the line (Perform PRINT-LINE)               *                     39
         *            Continue (using default queue manager name)       *                     40
         *         End-if                                               *                     41
         *                                                              *                     42
         *         If the name of the queue is missing                  *                     43
         *            Build an error message and move it into data line *                     44
         *            Print the line (Perform PRINT-LINE)               *                     45
         *            Branch to Exit2                                   *                     46
         *         End-if                                               *                     47
         *                                                              *                     48
         *      Print the rest of the header (Perform PRINT_HDR2)       *                     49

         COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE     2
         1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
         *                                                              *                     50
         *      Connect to the queue manager                            *                     51
         *      If an error occurs                                      *                     52
         *         Build an error message and move it into data line    *                     53
         *         Print the line (Perform PRINT-LINE)                  *                     54
         *         Branch to Exit2                                      *                     55
         *      End-if                                                  *                     56
         *                                                              *                     57
         *      Open the queue                                          *                     58
         *      If an error occurs                                      *                     59
         *         Build an error message and move it into data line    *                     60
         *         Print the line (Perform PRINT-LINE)                  *                     61
         *         Branch to Exit1                                      *                     62
         *      End-if                                                  *                     63
         *                                                              *                     64
         *      Get the first message (using BROWSE-FIRST option)       *                     65
         *                                                              *                     66
         *      Do while no error                                       *                     67
         *                                                              *                     68
         *         Add 1 to relative message number                     *                     69
```

Figure 22 MQGET Sample Compiled Listing (Page 1 of 35)

```
*         Move message into print line (maximum 80 bytes)     *                                70
*         Print the line (Perform PRINT-LINE)                 *                                71
*                                                             *                                72
*         Get next message (using BROWSE-NEXT option)         *                                73
*                                                             *                                74
*      End-do                                                 *                                75
*                                                             *                                76
*      When an error occurs                                   *                                77
*         If no more messages                                 *                                78
*            Do nothing                                       *                                79
*         else                                                *                                80
*            Build an error message and move it into data line *                               81
*            Print the line (Perform PRINT-LINE)              *                                82
*         End-if                                              *                                83
*      End-if                                                 *                                84
*                                                             *                                85
*      Close the queue                                        *                                86
*      If an error occurs                                     *                                87
*         Build an error message and move it into data line   *                                88
*         Print the line (Perform PRINT-LINE)                 *                                89
*      End-if                                                 *                                90
*                                                             *                                91
*   Exit1  (DISCONNECT)                                       *                                92
*   -----                                                     *                                93
*                                                             *                                94
*      Disconnect from the queue manager                      *                                95
*      If an error occurs                                     *                                96
*         Build an error message and move it into data line   *                                97
*         Print the line (Perform PRINT-LINE)                 *                                98

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE     3
1--------------------VISION:RESULTS FREE FORM TEXT----------------------72         --------
*      End-if                                                 *                                99
*                                                             *                               100
*   Exit2  (MAIN_END)                                         *                               101
*   -----                                                     *                               102
*                                                             *                               103
*      Set the return code                                    *                               104
*                                                             *                               105
*      Close print file                                       *                               106
*                                                             *                               107
*      Stop run                                               *                               108
*                                                             *                               109
*   Print line  (PRINT_LINE SECTION)                          *                               110
*   ----------                                                *                               111
*                                                             *                               112
*      If number of lines printed is greater than page maximum *                              113
*         Print first line of header (Perform PRINT_HDR1)     *                               114
*         Print the rest of the header (Perform PRINT_HDR2)*                                  115
*      End-if                                                 *                               116
*                                                             *                               117
*      Print data line                                        *                               118
*                                                             *                               119
*      Add 1 to count of lines printed                        *                               120
*                                                             *                               121
*      Return to performing section                           *                               122
*                                                             *                               123
*   Print first line of header  (PRINT_HDR1 SECTION)          *                               124
*   --------------------------                                *                               125
*                                                             *                               126
*      Add 1 to page number                                   *                               127
*                                                             *                               128
*      Print first line after jumping to top of page          *                               129
*                                                             *                               130
*      Set number of lines printed to 1                       *                               131
*                                                             *                               132
*      Return to performing section                           *                               133
*                                                             *                               134
*   Print rest of header  (PRINT_HDR2 SECTION)                *                               135
*   --------------------                                      *                               136
*                                                             *                               137
*      Print remaining header lines                           *                               138
*                                                             *                               139
*      Return to performing section                           *                               140
*                                                             *                               141
* ************************************************************* *                             142
* ------------------------------------------------------------- *                             143
FILE XFILE DUMMY                                                                              144
FILE SYSOUT FB 133 OUTPUT FROM SYSOUT                                                         145
PRINT_REC         133    CH                                                                   146
CARRIAGE_CONTROL    1  1 CH                                                                   147

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE     4
1--------------------VISION:RESULTS FREE FORM TEXT----------------------72         --------
PRINT_DATA        132  2 CH                                                                   148
```

Figure 22 MQGET Sample Compiled Listing (Page 2 of 35)

```
WORKAREA                                                                        149
  PARMLEN          2  BI VALUE  0                                               150
  LOW_VALUES      48  CH VALUE LOWVALUES                                        151
  MESSAGE_COUNT    4  PD                                                        152
  DATA_LENGTH      4  PD                                                        153
  COMP_CODE        4  PD                                                        154
  REASON_CODE      4  PD                                                        155
WORKAREA                                                                        156
  PARMINFO        60    VALUE NULL  ;JCL PARM Parameter                         157
  PARMDATA         1  1             ;PARM data one byte at a time              158
WORKAREA                                                                        159
      $COBOL                                                                    160
     * ------------------------------------------------------------ *           161
     *                                                                          162
     *    W00 - General work fields                                             163
     *                                                                          164
      01  W00-MAX-LINES              PIC S9(04) BINARY  VALUE +60.             165
      01  W00-LINE-COUNT             PIC S9(04) BINARY  VALUE ZERO.            166
      01  W00-PAGE-NUMBER            PIC S9(04) BINARY  VALUE ZERO.            167
      01  W00-MESSAGE-COUNT          PIC S9(09) BINARY  VALUE ZERO.            168
      01  W00-DATE.                                                            169
          05  W00-YY                 PIC X(02).                                170
          05  W00-MM                 PIC X(02).                                171
          05  W00-DD                 PIC X(02).                                172
      01  W00-PRINT-DATA             PIC X(132).                               173
     *                                                                          174
     *    W01 - Lines of the print report                                       175
     *                                                                          176
      01  W01-HEADER-1.                                                        177
          05  FILLER                 PIC X(10) VALUE SPACES.                   178
          05  W01-DATE               PIC X(10).                                179
          05  FILLER                 PIC X(36) VALUE SPACES.                   180
          05  FILLER                 PIC X(19) VALUE                           181
                                                  'SAMPLE QUEUE REPORT'.        182
          05  FILLER                 PIC X(38) VALUE SPACES.                   183
          05  FILLER                 PIC X(05) VALUE 'PAGE '.                  184
          05  W01-PAGE               PIC ZZZ9.                                 185
          05  FILLER                 PIC X(10) VALUE SPACES.                   186
      01  W01-HEADER-2.                                                        187
          05  FILLER                 PIC X(25) VALUE SPACES.                   188
          05  FILLER                 PIC X(29) VALUE                           189
                                     '       QUEUE MANAGER NAME : '.            190
          05  W01-MQM-NAME           PIC X(48) VALUE SPACES.                   191
          05  FILLER                 PIC X(30) VALUE SPACES.                   192
      01  W01-HEADER-3.                                                        193
          05  FILLER                 PIC X(37) VALUE SPACES.                   194
          05  FILLER                 PIC X(17) VALUE                           195
                                                    '   QUEUE NAME : '.       196

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE     5
1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
          05  W01-QUEUE-NAME         PIC X(48) VALUE SPACES.                   197
          05  FILLER                 PIC X(30) VALUE SPACES.                   198
      01  W01-HEADER-4.                                                        199
          05  FILLER                 PIC X(16)  VALUE SPACES.                  200
          05  FILLER                 PIC X(116) VALUE '  RELATIVE'.            201
      01  W01-HEADER-5.                                                        202
          05  FILLER                 PIC X(16)  VALUE SPACES.                  203
          05  FILLER                 PIC X(10)  VALUE '   MESSAGE'.            204
          05  FILLER                 PIC X(106) VALUE '   MESSAGE'.            205
      01  W01-HEADER-6.                                                        206
          05  FILLER                 PIC X(16) VALUE SPACES.                   207
          05  FILLER                 PIC X(10) VALUE '   NUMBER '.             208
          05  FILLER                 PIC X(10) VALUE '   LENGTH '.             209
          05  FILLER                 PIC X(96) VALUE                           210
                    '--------------------------------- MESSAGE DATA             211
     -              ' ---------------------------------'.                       212
      01  W01-REPORT-LINE.                                                     213
          05  FILLER                 PIC X(16) VALUE SPACES.                   214
          05  W01-MESSAGE-NUMBER     PIC Z(8)9.                                215
          05  FILLER                 PIC X     VALUE SPACE.                    216
          05  W01-MESSAGE-LENGTH     PIC Z(8)9.                                217
          05  FILLER                 PIC X     VALUE SPACE.                    218
          05  W01-DATA               PIC X(80).                                219
          05  FILLER                 PIC X(16) VALUE SPACES.                   220
     *                                                                          221
     *    W02 - Data fields derived from the PARM field                         222
     *                                                                          223
      01  W02-MQM                    PIC X(48) VALUE SPACES.                   224
      01  W02-OBJECT                 PIC X(48) VALUE SPACES.                   225
     *                                                                          226
     *    W04 - Error and information messages                                  227
     *                                                                          228
      01  W04-MESSAGE-0.                                                       229
          05  FILLER             PIC X(48) VALUE SPACES.                       230
```

Figure 22 MQGET Sample Compiled Listing (Page 3 of 35)

```
           05  FILLER             PIC X(35) VALUE                                  231
                                 '********** END OF REPORT **********'.           232
           05  FILLER             PIC X(49) VALUE SPACES.                          233
         01  W04-MESSAGE-1.                                                       234
           05  FILLER             PIC X(10)  VALUE SPACES.                         235
           05  FILLER             PIC X(122) VALUE                                 236
             '********** NO DATA PASSED TO PROGRAM. PROGRAM REQUIRES A             237
      -      'QUEUE MANAGER NAME AND A QUEUE NAME. **********'.                   238
         01  W04-MESSAGE-2.                                                       239
           05  FILLER             PIC X(25)  VALUE SPACES.                         240
           05  FILLER             PIC X(107) VALUE                                 241
             '********** NO QUEUE MANAGER NAME PASSED TO PROGRAM - DEFA             242
      -      'ULT USED *****'.                                                    243
         01  W04-MESSAGE-3.                                                       244
           05  FILLER             PIC X(38) VALUE SPACES.                          245

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE     6
1-------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
           05  FILLER             PIC X(94) VALUE                                  246
             '********** NO QUEUE NAME PASSED TO PROGRAM. **********'.             247
         01  W04-MESSAGE-4.                                                       248
           05  FILLER             PIC X(13) VALUE SPACES.                          249
           05  FILLER             PIC X(32) VALUE                                  250
                '********** AN ERROR OCCURRED IN '.                               251
           05  W04-MSG4-TYPE      PIC X(10).                                       252
           05  FILLER             PIC X(20) VALUE                                  253
                '. COMPLETION CODE = '.                                           254
           05  W04-MSG4-COMPCODE  PIC Z(8)9.                                       255
           05  FILLER             PIC X(15) VALUE ' REASON CODE ='.                256
           05  W04-MSG4-REASON    PIC Z(8)9.                                       257
           05  FILLER             PIC X(24) VALUE ' **********'.                   258
      *                                                                           259
      *    W06 - Return values                                                    260
      *                                                                           261
         01  W06-CSQ4-OK            PIC S9(4) VALUE 0.                           262
         01  W06-CSQ4-WARNING       PIC S9(4) VALUE 4.                           263
         01  W06-CSQ4-ERROR         PIC S9(4) VALUE 8.                           264
         $ECOBOL                                                                  265
      *                                                                           266
      *    The following copy files define API control blocks.                    267
      *                                                                           268
      WORKAREA                                                                    269
      COPY CMQWORK COBOL                                                          270
COPIED      ************************************************************        271
COPIED      **                                                        **        272
COPIED      **               WebSphere MQ for z/OS                    **        273
COPIED      **                                                        **        274
COPIED      **  FILE NAME:     CMQWORK                               **        275
COPIED      **                                                        **        276
COPIED      **  DESCRIPTION:   Work Area Structures And Parms        **        277
COPIED      **                                                        **        278
COPIED      ************************************************************        279
COPIED      **                                                        **        280
COPIED      **  FUNCTION:      This file declares work area structures and **    281
COPIED      **                 parms, which are not found in other files.  **    282
COPIED      **                                                        **        283
COPIED      **                 This file also contains a copy of the COBOL **    284
COPIED      **                 data file called CMQV which has an      **        285
COPIED      **                 assortment of constants used by the MQIs.   **    286
COPIED      **                 and other COBOL data structures.        **        287
COPIED      **                                                        **        288
COPIED      **  PROCESSOR:     COBOL                                 **        289
COPIED      **                                                        **        290
COPIED      ************************************************************        291
COPIED      * QUEUE MANAGER NAME                                                292
COPIED       01  QMGR            PIC X(48).                                     293
COPIED      * CONNECTION HANDLE                                                 294

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE     7
1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED       01  HCONN           PIC S9(9) BINARY.                              295
COPIED      * OBJECT HANDLE                                                     296
COPIED       01  HOBJ            PIC S9(9) BINARY.                              297
COPIED      * OPTIONS THAT CONTROL THE ACTION                                   298
COPIED       01  HOPTIONS        PIC S9(9) BINARY.                              299
COPIED      * COMPLETION CODE                                                   300
COPIED       01  COMPCODE        PIC S9(9) BINARY.                              301
COPIED      * REASON CODE                                                       302
COPIED       01  REASON          PIC S9(9) BINARY.                              303
COPIED      * TEXT ERROR MESSAGE                                                304
COPIED       01  ERRORMSG        PIC X(200).                                    305
COPIED      * TOTAL LENGTH OF MESSAGE BUFFER                                    306
COPIED       01  BUFFERLEN       PIC S9(9) BINARY.                              307
COPIED      * MESSAGE BUFFER                                                    308
COPIED       01  BUFFER.                                                        309
```

Figure 22 MQGET Sample Compiled Listing (Page 4 of 35)

```
COPIED          02  BUFFERDATA     PIC X(32767).                                  310
COPIED          02  BUFFERARRAY REDEFINES BUFFERDATA                              311
COPIED                          PIC X(1) OCCURS 32767 TIMES.                     312
COPIED       * LENGTH OF THE DATA IN THE MESSAGE BUFFER                          313
COPIED        01  DATALEN          PIC S9(9) BINARY.                            314
COPIED       * COUNT OF ATTRIBUTE SELECTORS                                      315
COPIED        01  SELECTORCOUNT     PIC S9(9) BINARY.                           316
COPIED       * ARRAY OF ATTRIBUTE SELECTORS                                      317
COPIED        01  SELECTORSTABLE.                                               318
COPIED           02  SELECTORS      PIC S9(9) BINARY OCCURS 10 TIMES.            319
COPIED       * COUNT OF INTEGER ATTRIBUTES                                       320
COPIED        01  INTATTRCOUNT PIC S9(9) BINARY.                                321
COPIED       * ARRAY OF INTEGER ATTRIBUTES                                       322
COPIED        01  INTATTRSTABLE.                                                323
COPIED           02  INTATTRS  PIC S9(9) BINARY OCCURS 10 TIMES.                 324
COPIED       * CHARACTER ATTRIBUTE BUFFER LENGTH                                 325
COPIED        01  CHARATTRLENGTH PIC S9(9) BINARY.                              326
COPIED       * CHARACTER ATTRIBUTES                                              327
COPIED        01  CHARATTRS.                                                    328
COPIED           02  CHARATTRSDATA  PIC X(32767).                               329
COPIED           02  CHARATTRSARRAY REDEFINES CHARATTRSDATA                     330
COPIED                          PIC X(1) OCCURS 32767 TIMES.                     331
COPIED       ******************************************************************* 332
COPIED       **  Values Related to MQCIH Structure                            ** 333
COPIED       ******************************************************************* 334
COPIED                                                                           335
COPIED       **   Structure Identifier                                           336
COPIED         10 MQCIH-STRUC-ID PIC X(4) VALUE 'CIH '.                          337
COPIED                                                                           338
COPIED       **   Structure Version Number                                       339
COPIED         10 MQCIH-VERSION-1       PIC S9(9) BINARY VALUE 1.               340
COPIED         10 MQCIH-VERSION-2       PIC S9(9) BINARY VALUE 2.               341
COPIED         10 MQCIH-CURRENT-VERSION PIC S9(9) BINARY VALUE 2.               342
COPIED                                                                           343

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE     8
      1--------------------VISION:RESULTS FREE FORM TEXT----------------------72         --------
COPIED       **   Structure Length                                               344
COPIED         10 MQCIH-LENGTH-1       PIC S9(9) BINARY VALUE 164.             345
COPIED         10 MQCIH-LENGTH-2       PIC S9(9) BINARY VALUE 180.             346
COPIED         10 MQCIH-CURRENT-LENGTH PIC S9(9) BINARY VALUE 180.             347
COPIED                                                                           348
COPIED       **   Flags                                                          349
COPIED         10 MQCIH-NONE PIC S9(9) BINARY VALUE 0.                           350
COPIED                                                                           351
COPIED       **   Return Code                                                    352
COPIED         10 MQCRC-OK                   PIC S9(9) BINARY VALUE 0.         353
COPIED         10 MQCRC-CICS-EXEC-ERROR      PIC S9(9) BINARY VALUE 1.         354
COPIED         10 MQCRC-MQ-API-ERROR         PIC S9(9) BINARY VALUE 2.         355
COPIED         10 MQCRC-BRIDGE-ERROR         PIC S9(9) BINARY VALUE 3.         356
COPIED         10 MQCRC-BRIDGE-ABEND         PIC S9(9) BINARY VALUE 4.         357
COPIED         10 MQCRC-APPLICATION-ABEND    PIC S9(9) BINARY VALUE 5.         358
COPIED         10 MQCRC-SECURITY-ERROR       PIC S9(9) BINARY VALUE 6.         359
COPIED         10 MQCRC-PROGRAM-NOT-AVAILABLE PIC S9(9) BINARY VALUE 7.        360
COPIED         10 MQCRC-BRIDGE-TIMEOUT       PIC S9(9) BINARY VALUE 8.         361
COPIED         10 MQCRC-TRANSID-NOT-AVAILABLE PIC S9(9) BINARY VALUE 9.        362
COPIED                                                                           363
COPIED       **   Unit of Work Control                                           364
COPIED         10 MQCUOWC-ONLY     PIC S9(9) BINARY VALUE 273.                   365
COPIED         10 MQCUOWC-CONTINUE PIC S9(9) BINARY VALUE 65536.                 366
COPIED         10 MQCUOWC-FIRST    PIC S9(9) BINARY VALUE 17.                    367
COPIED         10 MQCUOWC-MIDDLE   PIC S9(9) BINARY VALUE 16.                    368
COPIED         10 MQCUOWC-LAST     PIC S9(9) BINARY VALUE 272.                   369
COPIED         10 MQCUOWC-COMMIT   PIC S9(9) BINARY VALUE 256.                   370
COPIED         10 MQCUOWC-BACKOUT  PIC S9(9) BINARY VALUE 4352.                  371
COPIED                                                                           372
COPIED       **   Get Wait Interval                                              373
COPIED         10 MQOGWI-DEFAULT PIC S9(9) BINARY VALUE -2.                      374
COPIED                                                                           375
COPIED       **   Link Type                                                      376
COPIED         10 MQCLT-PROGRAM     PIC S9(9) BINARY VALUE 1.                   377
COPIED         10 MQCLT-TRANSACTION PIC S9(9) BINARY VALUE 2.                   378
COPIED                                                                           379
COPIED       **   Output Data Length                                             380
COPIED         10 MQCODL-AS-INPUT PIC S9(9) BINARY VALUE -1.                     381
COPIED                                                                           382
COPIED       **   ADS Descriptor                                                 383
COPIED         10 MQCADSD-NONE      PIC S9(9) BINARY VALUE 0.                   384
COPIED         10 MQCADSD-SEND      PIC S9(9) BINARY VALUE 1.                   385
COPIED         10 MQCADSD-RECV      PIC S9(9) BINARY VALUE 16.                  386
COPIED         10 MQCADSD-MSGFORMAT PIC S9(9) BINARY VALUE 256.                 387
COPIED                                                                           388
COPIED       **   Conversational Task                                            389
COPIED         10 MQCCT-YES PIC S9(9) BINARY VALUE 1.                           390
COPIED         10 MQCCT-NO  PIC S9(9) BINARY VALUE 0.                           391
```

Figure 22 MQGET Sample Compiled Listing (Page 5 of 35)

```
COPIED                                                                          392

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE     9
      1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED      **   Task End Status                                                    393
COPIED        10 MQCTES-NOSYNC  PIC S9(9) BINARY VALUE 0.                         394
COPIED        10 MQCTES-COMMIT  PIC S9(9) BINARY VALUE 256.                       395
COPIED        10 MQCTES-BACKOUT PIC S9(9) BINARY VALUE 4352.                      396
COPIED        10 MQCTES-ENDTASK PIC S9(9) BINARY VALUE 65536.                     397
COPIED                                                                          398
COPIED      **   Facility                                                       399
COPIED        10 MQCFAC-NONE PIC X(8) VALUE LOW-VALUES.                          400
COPIED                                                                          401
COPIED      **   Function                                                       402
COPIED        10 MQCFUNC-MQCONN PIC X(4) VALUE 'CONN'.                           403
COPIED        10 MQCFUNC-MQGET  PIC X(4) VALUE 'GET '.                           404
COPIED        10 MQCFUNC-MQINQ  PIC X(4) VALUE 'INQ '.                           405
COPIED        10 MQCFUNC-MQOPEN PIC X(4) VALUE 'OPEN'.                           406
COPIED        10 MQCFUNC-MQPUT  PIC X(4) VALUE 'PUT '.                           407
COPIED        10 MQCFUNC-MQPUT1 PIC X(4) VALUE 'PUT1'.                           408
COPIED        10 MQCFUNC-NONE   PIC X(4) VALUE SPACES.                           409
COPIED                                                                          410
COPIED      **   Start Code                                                     411
COPIED        10 MQCSC-START     PIC X(4) VALUE 'S   '.                          412
COPIED        10 MQCSC-STARTDATA PIC X(4) VALUE 'SD  '.                          413
COPIED        10 MQCSC-TERMINPUT PIC X(4) VALUE 'TD  '.                          414
COPIED        10 MQCSC-NONE      PIC X(4) VALUE SPACES.                          415
COPIED                                                                          416
COPIED                                                                          417
COPIED      ******************************************************************  418
COPIED      **  Values Related to MQCNO Structure                           **  419
COPIED      ******************************************************************  420
COPIED                                                                          421
COPIED      **   Structure Identifier                                           422
COPIED        10 MQCNO-STRUC-ID PIC X(4) VALUE 'CNO '.                          423
COPIED                                                                          424
COPIED      **   Structure Version Number                                       425
COPIED        10 MQCNO-VERSION-1       PIC S9(9) BINARY VALUE 1.              426
COPIED        10 MQCNO-VERSION-2       PIC S9(9) BINARY VALUE 2.              427
COPIED        10 MQCNO-VERSION-3       PIC S9(9) BINARY VALUE 3.              428
COPIED        10 MQCNO-CURRENT-VERSION PIC S9(9) BINARY VALUE 3.              429
COPIED                                                                          430
COPIED      **   Connect Options                                                431
COPIED        10 MQCNO-STANDARD-BINDING         PIC S9(9) BINARY VALUE 0.     432
COPIED        10 MQCNO-FASTPATH-BINDING         PIC S9(9) BINARY VALUE 1.     433
COPIED        10 MQCNO-SERIALIZE-CONN-TAG-Q-MGR PIC S9(9) BINARY VALUE 2.     434
COPIED        10 MQCNO-SERIALIZE-CONN-TAG-QSG   PIC S9(9) BINARY VALUE 4.     435
COPIED        10 MQCNO-RESTRICT-CONN-TAG-Q-MGR  PIC S9(9) BINARY VALUE 8.     436
COPIED        10 MQCNO-RESTRICT-CONN-TAG-QSG    PIC S9(9) BINARY VALUE 16.    437
COPIED        10 MQCNO-NONE                     PIC S9(9) BINARY VALUE 0.     438
COPIED                                                                          439
COPIED      **   Queue-Manager Connection Tag                                   440
COPIED        10 MQCT-NONE PIC X(128) VALUE LOW-VALUES.                         441

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    10
      1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED                                                                          442
COPIED                                                                          443
COPIED      ******************************************************************  444
COPIED      **  Values Related to MQDLH Structure                           **  445
COPIED      ******************************************************************  446
COPIED                                                                          447
COPIED      **   Structure Identifier                                           448
COPIED        10 MQDLH-STRUC-ID PIC X(4) VALUE 'DLH '.                          449
COPIED                                                                          450
COPIED      **   Structure Version Number                                       451
COPIED        10 MQDLH-VERSION-1       PIC S9(9) BINARY VALUE 1.              452
COPIED        10 MQDLH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.              453
COPIED                                                                          454
COPIED                                                                          455
COPIED      ******************************************************************  456
COPIED      **  Values Related to MQGMO Structure                           **  457
COPIED      ******************************************************************  458
COPIED                                                                          459
COPIED      **   Structure Identifier                                           460
COPIED        10 MQGMO-STRUC-ID PIC X(4) VALUE 'GMO '.                          461
COPIED                                                                          462
COPIED      **   Structure Version Number                                       463
COPIED        10 MQGMO-VERSION-1       PIC S9(9) BINARY VALUE 1.              464
COPIED        10 MQGMO-VERSION-2       PIC S9(9) BINARY VALUE 2.              465
COPIED        10 MQGMO-VERSION-3       PIC S9(9) BINARY VALUE 3.              466
COPIED        10 MQGMO-CURRENT-VERSION PIC S9(9) BINARY VALUE 3.              467
COPIED                                                                          468
COPIED      **   Get-Message Options                                            469
COPIED        10 MQGMO-WAIT                  PIC S9(9) BINARY VALUE 1.       470
```

Figure 22 MQGET Sample Compiled Listing (Page 6 of 35)

```
COPIED        10 MQGMO-NO-WAIT                 PIC S9(9) BINARY VALUE 0.                                  471
COPIED        10 MQGMO-SET-SIGNAL              PIC S9(9) BINARY VALUE 8.                                  472
COPIED        10 MQGMO-FAIL-IF-QUIESCING       PIC S9(9) BINARY VALUE 8192.                               473
COPIED        10 MQGMO-SYNCPOINT               PIC S9(9) BINARY VALUE 2.                                  474
COPIED        10 MQGMO-SYNCPOINT-IF-PERSISTENT PIC S9(9) BINARY VALUE 4096.                               475
COPIED        10 MQGMO-NO-SYNCPOINT            PIC S9(9) BINARY VALUE 4.                                  476
COPIED        10 MQGMO-MARK-SKIP-BACKOUT       PIC S9(9) BINARY VALUE 128.                                477
COPIED        10 MQGMO-BROWSE-FIRST            PIC S9(9) BINARY VALUE 16.                                 478
COPIED        10 MQGMO-BROWSE-NEXT             PIC S9(9) BINARY VALUE 32.                                 479
COPIED        10 MQGMO-BROWSE-MSG-UNDER-CURSOR PIC S9(9) BINARY VALUE 2048.                               480
COPIED        10 MQGMO-MSG-UNDER-CURSOR        PIC S9(9) BINARY VALUE 256.                                481
COPIED        10 MQGMO-LOCK                    PIC S9(9) BINARY VALUE 512.                                482
COPIED        10 MQGMO-UNLOCK                  PIC S9(9) BINARY VALUE 1024.                               483
COPIED        10 MQGMO-ACCEPT-TRUNCATED-MSG    PIC S9(9) BINARY VALUE 64.                                 484
COPIED        10 MQGMO-CONVERT                 PIC S9(9) BINARY VALUE 16384.                              485
COPIED        10 MQGMO-LOGICAL-ORDER           PIC S9(9) BINARY VALUE 32768.                              486
COPIED        10 MQGMO-COMPLETE-MSG            PIC S9(9) BINARY VALUE 65536.                              487
COPIED        10 MQGMO-ALL-MSGS-AVAILABLE      PIC S9(9) BINARY VALUE 131072.                             488
COPIED        10 MQGMO-ALL-SEGMENTS-AVAILABLE  PIC S9(9) BINARY VALUE 262144.                             489
COPIED        10 MQGMO-NONE                    PIC S9(9) BINARY VALUE 0.                                  490

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    11
      1--------------------VISION:RESULTS FREE FORM TEXT----------------------72          --------
COPIED                                                                                                    491
COPIED      **   Wait Interval                                                                            492
COPIED        10 MQWI-UNLIMITED PIC S9(9) BINARY VALUE -1.                                               493
COPIED                                                                                                    494
COPIED      **   Signal Values                                                                            495
COPIED        10 MQEC-MSG-ARRIVED           PIC S9(9) BINARY VALUE 2.                                   496
COPIED        10 MQEC-WAIT-INTERVAL-EXPIRED PIC S9(9) BINARY VALUE 3.                                   497
COPIED        10 MQEC-WAIT-CANCELED         PIC S9(9) BINARY VALUE 4.                                   498
COPIED        10 MQEC-Q-MGR-QUIESCING       PIC S9(9) BINARY VALUE 5.                                   499
COPIED        10 MQEC-CONNECTION-QUIESCING  PIC S9(9) BINARY VALUE 6.                                   500
COPIED                                                                                                    501
COPIED      **   Match Options                                                                            502
COPIED        10 MQMO-MATCH-MSG-ID         PIC S9(9) BINARY VALUE 1.                                    503
COPIED        10 MQMO-MATCH-CORREL-ID      PIC S9(9) BINARY VALUE 2.                                    504
COPIED        10 MQMO-MATCH-GROUP-ID       PIC S9(9) BINARY VALUE 4.                                    505
COPIED        10 MQMO-MATCH-MSG-SEQ-NUMBER PIC S9(9) BINARY VALUE 8.                                    506
COPIED        10 MQMO-MATCH-MSG-TOKEN      PIC S9(9) BINARY VALUE 32.                                   507
COPIED        10 MQMO-NONE                 PIC S9(9) BINARY VALUE 0.                                    508
COPIED                                                                                                    509
COPIED      **   Group Status                                                                             510
COPIED        10 MQGS-NOT-IN-GROUP      PIC X VALUE ' '.                                                511
COPIED        10 MQGS-MSG-IN-GROUP      PIC X VALUE 'G'.                                                512
COPIED        10 MQGS-LAST-MSG-IN-GROUP PIC X VALUE 'L'.                                                513
COPIED                                                                                                    514
COPIED      **   Segment Status                                                                           515
COPIED        10 MQSS-NOT-A-SEGMENT PIC X VALUE ' '.                                                    516
COPIED        10 MQSS-SEGMENT       PIC X VALUE 'S'.                                                    517
COPIED        10 MQSS-LAST-SEGMENT  PIC X VALUE 'L'.                                                    518
COPIED                                                                                                    519
COPIED      **   Segmentation                                                                             520
COPIED        10 MQSEG-INHIBITED PIC X VALUE ' '.                                                       521
COPIED        10 MQSEG-ALLOWED   PIC X VALUE 'A'.                                                       522
COPIED                                                                                                    523
COPIED      **   Message Token                                                                            524
COPIED        10 MQMTOK-NONE PIC X(16) VALUE LOW-VALUES.                                                525
COPIED                                                                                                    526
COPIED      **   Returned Length                                                                          527
COPIED        10 MQRL-UNDEFINED PIC S9(9) BINARY VALUE -1.                                              528
COPIED                                                                                                    529
COPIED                                                                                                    530
COPIED      ***************************************************************                           531
COPIED      **  Values Related to MQIIH Structure                          **                           532
COPIED      ***************************************************************                           533
COPIED                                                                                                    534
COPIED      **   Structure Identifier                                                                     535
COPIED        10 MQIIH-STRUC-ID PIC X(4) VALUE 'IIH '.                                                  536
COPIED                                                                                                    537
COPIED      **   Structure Version Number                                                                 538
COPIED        10 MQIIH-VERSION-1       PIC S9(9) BINARY VALUE 1.                                        539

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    12
      1--------------------VISION:RESULTS FREE FORM TEXT----------------------72          --------
COPIED        10 MQIIH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.                                        540
COPIED                                                                                                    541
COPIED      **   Structure Length                                                                         542
COPIED        10 MQIIH-LENGTH-1 PIC S9(9) BINARY VALUE 84.                                              543
COPIED                                                                                                    544
COPIED      **   Flags                                                                                    545
COPIED        10 MQIIH-NONE PIC S9(9) BINARY VALUE 0.                                                   546
COPIED                                                                                                    547
COPIED      **   Authenticator                                                                            548
COPIED        10 MQIAUT-NONE PIC X(8) VALUE SPACES.                                                     549
```

Figure 22 MQGET Sample Compiled Listing (Page 7 of 35)

```
COPIED                                                                          550
COPIED       **   Transaction Instance Identifier                               551
COPIED         10 MQITII-NONE PIC X(16) VALUE LOW-VALUES.                       552
COPIED                                                                          553
COPIED       **   Transaction State                                             554
COPIED         10 MQITS-IN-CONVERSATION     PIC X VALUE 'C'.                    555
COPIED         10 MQITS-NOT-IN-CONVERSATION PIC X VALUE ' '.                    556
COPIED         10 MQITS-ARCHITECTED         PIC X VALUE 'A'.                    557
COPIED                                                                          558
COPIED       **   Commit Mode                                                   559
COPIED         10 MQICM-COMMIT-THEN-SEND PIC X VALUE '0'.                       560
COPIED         10 MQICM-SEND-THEN-COMMIT PIC X VALUE '1'.                       561
COPIED                                                                          562
COPIED       **   Security Scope                                                563
COPIED         10 MQISS-CHECK PIC X VALUE 'C'.                                  564
COPIED         10 MQISS-FULL  PIC X VALUE 'F'.                                  565
COPIED                                                                          566
COPIED                                                                          567
COPIED       ******************************************************************568
COPIED       **  Values Related to MQMD Structure                            **569
COPIED       ******************************************************************570
COPIED                                                                          571
COPIED       **   Structure Identifier                                          572
COPIED         10 MQMD-STRUC-ID PIC X(4) VALUE 'MD  '.                          573
COPIED                                                                          574
COPIED       **   Structure Version Number                                      575
COPIED         10 MQMD-VERSION-1       PIC S9(9) BINARY VALUE 1.                576
COPIED         10 MQMD-VERSION-2       PIC S9(9) BINARY VALUE 2.                577
COPIED         10 MQMD-CURRENT-VERSION PIC S9(9) BINARY VALUE 2.                578
COPIED                                                                          579
COPIED       **   Report Options                                                580
COPIED         10 MQRO-EXCEPTION                PIC S9(9) BINARY VALUE         581
COPIED                                                        16777216.         582
COPIED         10 MQRO-EXCEPTION-WITH-DATA      PIC S9(9) BINARY VALUE         583
COPIED                                                        50331648.         584
COPIED         10 MQRO-EXCEPTION-WITH-FULL-DATA PIC S9(9) BINARY VALUE         585
COPIED                                                       117440512.         586
COPIED         10 MQRO-EXPIRATION               PIC S9(9) BINARY VALUE         587
COPIED                                                         2097152.         588

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    13
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED         10 MQRO-EXPIRATION-WITH-DATA     PIC S9(9) BINARY VALUE         589
COPIED                                                          6291456.         590
COPIED         10 MQRO-EXPIRATION-WITH-FULL-DATA PIC S9(9) BINARY VALUE        591
COPIED                                                         14680064.        592
COPIED         10 MQRO-COA                      PIC S9(9) BINARY VALUE 256.    593
COPIED         10 MQRO-COA-WITH-DATA            PIC S9(9) BINARY VALUE 768.    594
COPIED         10 MQRO-COA-WITH-FULL-DATA       PIC S9(9) BINARY VALUE 1792.   595
COPIED         10 MQRO-COD                      PIC S9(9) BINARY VALUE 2048.   596
COPIED         10 MQRO-COD-WITH-DATA            PIC S9(9) BINARY VALUE 6144.   597
COPIED         10 MQRO-COD-WITH-FULL-DATA       PIC S9(9) BINARY VALUE 14336.  598
COPIED         10 MQRO-PAN                      PIC S9(9) BINARY VALUE 1.      599
COPIED         10 MQRO-NAN                      PIC S9(9) BINARY VALUE 2.      600
COPIED         10 MQRO-NEW-MSG-ID               PIC S9(9) BINARY VALUE 0.      601
COPIED         10 MQRO-PASS-MSG-ID              PIC S9(9) BINARY VALUE 128.    602
COPIED         10 MQRO-COPY-MSG-ID-TO-CORREL-ID PIC S9(9) BINARY VALUE 0.      603
COPIED         10 MQRO-PASS-CORREL-ID           PIC S9(9) BINARY VALUE 64.     604
COPIED         10 MQRO-DEAD-LETTER-Q            PIC S9(9) BINARY VALUE 0.      605
COPIED         10 MQRO-DISCARD-MSG              PIC S9(9) BINARY VALUE         606
COPIED                                                        134217728.        607
COPIED         10 MQRO-NONE                     PIC S9(9) BINARY VALUE 0.      608
COPIED                                                                         609
COPIED       **   Report Options Masks                                          610
COPIED         10 MQRO-REJECT-UNSUP-MASK        PIC S9(9) BINARY VALUE         611
COPIED                                                        270270464.        612
COPIED         10 MQRO-ACCEPT-UNSUP-MASK        PIC S9(9) BINARY VALUE         613
COPIED                                                       -270532353.        614
COPIED         10 MQRO-ACCEPT-UNSUP-IF-XMIT-MASK PIC S9(9) BINARY VALUE 261888. 615
COPIED                                                                         616
COPIED       **   Message Types                                                 617
COPIED         10 MQMT-SYSTEM-FIRST       PIC S9(9) BINARY VALUE 1.           618
COPIED         10 MQMT-REQUEST            PIC S9(9) BINARY VALUE 1.           619
COPIED         10 MQMT-REPLY              PIC S9(9) BINARY VALUE 2.           620
COPIED         10 MQMT-DATAGRAM           PIC S9(9) BINARY VALUE 8.           621
COPIED         10 MQMT-REPORT             PIC S9(9) BINARY VALUE 4.           622
COPIED         10 MQMT-MQE-FIELDS-FROM-MQE PIC S9(9) BINARY VALUE 112.        623
COPIED         10 MQMT-MQE-FIELDS         PIC S9(9) BINARY VALUE 113.         624
COPIED         10 MQMT-SYSTEM-LAST        PIC S9(9) BINARY VALUE 65535.       625
COPIED         10 MQMT-APPL-FIRST         PIC S9(9) BINARY VALUE 65536.       626
COPIED         10 MQMT-APPL-LAST          PIC S9(9) BINARY VALUE 999999999.   627
COPIED                                                                         628
COPIED       **   Expiry                                                        629
COPIED         10 MQEI-UNLIMITED PIC S9(9) BINARY VALUE -1.                     630
COPIED                                                                         631
```

Figure 22 MQGET Sample Compiled Listing (Page 8 of 35)

```
COPIED      **   Feedback Values                                                    632
COPIED        10 MQFB-NONE                  PIC S9(9) BINARY VALUE 0.              633
COPIED        10 MQFB-SYSTEM-FIRST          PIC S9(9) BINARY VALUE 1.              634
COPIED        10 MQFB-QUIT                  PIC S9(9) BINARY VALUE 256.            635
COPIED        10 MQFB-EXPIRATION            PIC S9(9) BINARY VALUE 258.            636
COPIED        10 MQFB-COA                   PIC S9(9) BINARY VALUE 259.            637

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    14
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQFB-COD                   PIC S9(9) BINARY VALUE 260.            638
COPIED        10 MQFB-CHANNEL-COMPLETED     PIC S9(9) BINARY VALUE 262.            639
COPIED        10 MQFB-CHANNEL-FAIL-RETRY    PIC S9(9) BINARY VALUE 263.            640
COPIED        10 MQFB-CHANNEL-FAIL          PIC S9(9) BINARY VALUE 264.            641
COPIED        10 MQFB-APPL-CANNOT-BE-STARTED PIC S9(9) BINARY VALUE 265.           642
COPIED        10 MQFB-TM-ERROR              PIC S9(9) BINARY VALUE 266.            643
COPIED        10 MQFB-APPL-TYPE-ERROR       PIC S9(9) BINARY VALUE 267.            644
COPIED        10 MQFB-STOPPED-BY-MSG-EXIT   PIC S9(9) BINARY VALUE 268.            645
COPIED        10 MQFB-XMIT-Q-MSG-ERROR      PIC S9(9) BINARY VALUE 271.            646
COPIED        10 MQFB-PAN                   PIC S9(9) BINARY VALUE 275.            647
COPIED        10 MQFB-NAN                   PIC S9(9) BINARY VALUE 276.            648
COPIED        10 MQFB-STOPPED-BY-CHAD-EXIT  PIC S9(9) BINARY VALUE 277.            649
COPIED        10 MQFB-STOPPED-BY-PUBSUB-EXIT PIC S9(9) BINARY VALUE 279.           650
COPIED        10 MQFB-NOT-A-REPOSITORY-MSG  PIC S9(9) BINARY VALUE 280.            651
COPIED        10 MQFB-BIND-OPEN-CLUSRCVR-DEL PIC S9(9) BINARY VALUE 281.           652
COPIED        10 MQFB-DATA-LENGTH-ZERO      PIC S9(9) BINARY VALUE 291.            653
COPIED        10 MQFB-DATA-LENGTH-NEGATIVE  PIC S9(9) BINARY VALUE 292.            654
COPIED        10 MQFB-DATA-LENGTH-TOO-BIG   PIC S9(9) BINARY VALUE 293.            655
COPIED        10 MQFB-BUFFER-OVERFLOW       PIC S9(9) BINARY VALUE 294.            656
COPIED        10 MQFB-LENGTH-OFF-BY-ONE     PIC S9(9) BINARY VALUE 295.            657
COPIED        10 MQFB-IIH-ERROR             PIC S9(9) BINARY VALUE 296.            658
COPIED        10 MQFB-NOT-AUTHORIZED-FOR-IMS PIC S9(9) BINARY VALUE 298.           659
COPIED        10 MQFB-IMS-ERROR             PIC S9(9) BINARY VALUE 300.            660
COPIED        10 MQFB-IMS-FIRST             PIC S9(9) BINARY VALUE 301.            661
COPIED        10 MQFB-IMS-LAST              PIC S9(9) BINARY VALUE 399.            662
COPIED        10 MQFB-CICS-INTERNAL-ERROR   PIC S9(9) BINARY VALUE 401.            663
COPIED        10 MQFB-CICS-NOT-AUTHORIZED   PIC S9(9) BINARY VALUE 402.            664
COPIED        10 MQFB-CICS-BRIDGE-FAILURE   PIC S9(9) BINARY VALUE 403.            665
COPIED        10 MQFB-CICS-CORREL-ID-ERROR  PIC S9(9) BINARY VALUE 404.            666
COPIED        10 MQFB-CICS-CCSID-ERROR      PIC S9(9) BINARY VALUE 405.            667
COPIED        10 MQFB-CICS-ENCODING-ERROR   PIC S9(9) BINARY VALUE 406.            668
COPIED        10 MQFB-CICS-CIH-ERROR        PIC S9(9) BINARY VALUE 407.            669
COPIED        10 MQFB-CICS-UOW-ERROR        PIC S9(9) BINARY VALUE 408.            670
COPIED        10 MQFB-CICS-COMMAREA-ERROR   PIC S9(9) BINARY VALUE 409.            671
COPIED        10 MQFB-CICS-APPL-NOT-STARTED PIC S9(9) BINARY VALUE 410.            672
COPIED        10 MQFB-CICS-APPL-ABENDED     PIC S9(9) BINARY VALUE 411.            673
COPIED        10 MQFB-CICS-DLQ-ERROR        PIC S9(9) BINARY VALUE 412.            674
COPIED        10 MQFB-CICS-UOW-BACKED-OUT   PIC S9(9) BINARY VALUE 413.            675
COPIED        10 MQFB-SYSTEM-LAST           PIC S9(9) BINARY VALUE 65535.          676
COPIED        10 MQFB-APPL-FIRST            PIC S9(9) BINARY VALUE 65536.          677
COPIED        10 MQFB-APPL-LAST             PIC S9(9) BINARY VALUE 999999999.      678
COPIED                                                                             679
COPIED      **   Encoding                                                          680
COPIED        10 MQENC-NATIVE PIC S9(9) BINARY VALUE 785.                          681
COPIED                                                                             682
COPIED      **   Encoding Masks                                                    683
COPIED        10 MQENC-INTEGER-MASK  PIC S9(9) BINARY VALUE 15.                    684
COPIED        10 MQENC-DECIMAL-MASK  PIC S9(9) BINARY VALUE 240.                   685
COPIED        10 MQENC-FLOAT-MASK    PIC S9(9) BINARY VALUE 3840.                  686

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    15
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQENC-RESERVED-MASK PIC S9(9) BINARY VALUE -4096.                 687
COPIED                                                                             688
COPIED      **   Encodings for Binary Integers                                     689
COPIED        10 MQENC-INTEGER-UNDEFINED PIC S9(9) BINARY VALUE 0.                 690
COPIED        10 MQENC-INTEGER-NORMAL    PIC S9(9) BINARY VALUE 1.                 691
COPIED        10 MQENC-INTEGER-REVERSED  PIC S9(9) BINARY VALUE 2.                 692
COPIED                                                                             693
COPIED      **   Encodings for Packed-Decimal Integers                             694
COPIED        10 MQENC-DECIMAL-UNDEFINED PIC S9(9) BINARY VALUE 0.                 695
COPIED        10 MQENC-DECIMAL-NORMAL    PIC S9(9) BINARY VALUE 16.                696
COPIED        10 MQENC-DECIMAL-REVERSED  PIC S9(9) BINARY VALUE 32.                697
COPIED                                                                             698
COPIED      **   Encodings for Floating-Point Numbers                              699
COPIED        10 MQENC-FLOAT-UNDEFINED     PIC S9(9) BINARY VALUE 0.             700
COPIED        10 MQENC-FLOAT-IEEE-NORMAL   PIC S9(9) BINARY VALUE 256.           701
COPIED        10 MQENC-FLOAT-IEEE-REVERSED PIC S9(9) BINARY VALUE 512.           702
COPIED        10 MQENC-FLOAT-S390          PIC S9(9) BINARY VALUE 768.           703
COPIED                                                                             704
COPIED      **   Coded Character-Set Identifiers                                   705
COPIED        10 MQCCSI-UNDEFINED PIC S9(9) BINARY VALUE 0.                        706
COPIED        10 MQCCSI-DEFAULT   PIC S9(9) BINARY VALUE 0.                        707
COPIED        10 MQCCSI-Q-MGR     PIC S9(9) BINARY VALUE 0.                        708
COPIED        10 MQCCSI-INHERIT   PIC S9(9) BINARY VALUE -2.                       709
COPIED        10 MQCCSI-EMBEDDED  PIC S9(9) BINARY VALUE -1.                       710
```

Figure 22 MQGET Sample Compiled Listing (Page 9 of 35)

```
COPIED                                                                          711
COPIED      **   Formats                                                        712
COPIED        10 MQFMT-NONE              PIC X(8) VALUE SPACES.                713
COPIED        10 MQFMT-ADMIN             PIC X(8) VALUE 'MQADMIN '.            714
COPIED        10 MQFMT-CHANNEL-COMPLETED PIC X(8) VALUE 'MQCHCOM '.            715
COPIED        10 MQFMT-CICS              PIC X(8) VALUE 'MQCICS  '.            716
COPIED        10 MQFMT-COMMAND-1         PIC X(8) VALUE 'MQCMD1  '.            717
COPIED        10 MQFMT-COMMAND-2         PIC X(8) VALUE 'MQCMD2  '.            718
COPIED        10 MQFMT-DEAD-LETTER-HEADER PIC X(8) VALUE 'MQDEAD  '.           719
COPIED        10 MQFMT-EVENT             PIC X(8) VALUE 'MQEVENT '.            720
COPIED        10 MQFMT-IMS               PIC X(8) VALUE 'MQIMS   '.            721
COPIED        10 MQFMT-IMS-VAR-STRING    PIC X(8) VALUE 'MQIMSVS '.            722
COPIED        10 MQFMT-MD-EXTENSION      PIC X(8) VALUE 'MQHMDE  '.            723
COPIED        10 MQFMT-PCF               PIC X(8) VALUE 'MQPCF   '.            724
COPIED        10 MQFMT-REF-MSG-HEADER    PIC X(8) VALUE 'MQHREF  '.            725
COPIED        10 MQFMT-RF-HEADER         PIC X(8) VALUE 'MQHRF   '.            726
COPIED        10 MQFMT-RF-HEADER-2       PIC X(8) VALUE 'MQHRF2  '.            727
COPIED        10 MQFMT-STRING            PIC X(8) VALUE 'MQSTR   '.            728
COPIED        10 MQFMT-TRIGGER           PIC X(8) VALUE 'MQTRIG  '.            729
COPIED        10 MQFMT-WORK-INFO-HEADER  PIC X(8) VALUE 'MQHWIH  '.            730
COPIED        10 MQFMT-XMIT-Q-HEADER     PIC X(8) VALUE 'MQXMIT  '.            731
COPIED                                                                          732
COPIED      **   Priority                                                       733
COPIED        10 MQPRI-PRIORITY-AS-Q-DEF PIC S9(9) BINARY VALUE -1.            734
COPIED                                                                          735

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    16
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED      **   Persistence Values                                             736
COPIED        10 MQPER-NOT-PERSISTENT      PIC S9(9) BINARY VALUE 0.           737
COPIED        10 MQPER-PERSISTENT          PIC S9(9) BINARY VALUE 1.           738
COPIED        10 MQPER-PERSISTENCE-AS-Q-DEF PIC S9(9) BINARY VALUE 2.          739
COPIED                                                                          740
COPIED      **   Message Identifier                                             741
COPIED        10 MQMI-NONE PIC X(24) VALUE LOW-VALUES.                         742
COPIED                                                                          743
COPIED      **   Correlation Identifier                                         744
COPIED        10 MQCI-NONE       PIC X(24) VALUE LOW-VALUES.                   745
COPIED        10 MQCI-NEW-SESSION.                                             746
COPIED           15 MQCI-NEW-SESSION1  PIC X(9) VALUE                          747
COPIED                X'414D51214E45575F53'.                                    748
COPIED           15 MQCI-NEW-SESSION2  PIC X(9) VALUE                          749
COPIED                X'455353494F4E5F434F'.                                    750
COPIED           15 MQCI-NEW-SESSION3  PIC X(6) VALUE                          751
COPIED                X'5252454C4944'.                                          752
COPIED                                                                          753
COPIED      **   Accounting Token                                               754
COPIED        10 MQACT-NONE PIC X(32) VALUE LOW-VALUES.                        755
COPIED                                                                          756
COPIED      **   Put Application Type                                           757
COPIED        10 MQAT-UNKNOWN          PIC S9(9) BINARY VALUE -1.            758
COPIED        10 MQAT-NO-CONTEXT       PIC S9(9) BINARY VALUE 0.             759
COPIED        10 MQAT-CICS             PIC S9(9) BINARY VALUE 1.             760
COPIED        10 MQAT-MVS              PIC S9(9) BINARY VALUE 2.             761
COPIED        10 MQAT-OS390            PIC S9(9) BINARY VALUE 2.             762
COPIED        10 MQAT-ZOS              PIC S9(9) BINARY VALUE 2.             763
COPIED        10 MQAT-IMS              PIC S9(9) BINARY VALUE 3.             764
COPIED        10 MQAT-OS2              PIC S9(9) BINARY VALUE 4.             765
COPIED        10 MQAT-DOS              PIC S9(9) BINARY VALUE 5.             766
COPIED        10 MQAT-AIX              PIC S9(9) BINARY VALUE 6.             767
COPIED        10 MQAT-UNIX             PIC S9(9) BINARY VALUE 6.             768
COPIED        10 MQAT-QMGR             PIC S9(9) BINARY VALUE 7.             769
COPIED        10 MQAT-OS400            PIC S9(9) BINARY VALUE 8.             770
COPIED        10 MQAT-WINDOWS          PIC S9(9) BINARY VALUE 9.             771
COPIED        10 MQAT-CICS-VSE         PIC S9(9) BINARY VALUE 10.            772
COPIED        10 MQAT-WINDOWS-NT       PIC S9(9) BINARY VALUE 11.            773
COPIED        10 MQAT-VMS              PIC S9(9) BINARY VALUE 12.            774
COPIED        10 MQAT-GUARDIAN         PIC S9(9) BINARY VALUE 13.            775
COPIED        10 MQAT-NSK              PIC S9(9) BINARY VALUE 13.            776
COPIED        10 MQAT-VOS              PIC S9(9) BINARY VALUE 14.            777
COPIED        10 MQAT-IMS-BRIDGE       PIC S9(9) BINARY VALUE 19.            778
COPIED        10 MQAT-XCF              PIC S9(9) BINARY VALUE 20.            779
COPIED        10 MQAT-CICS-BRIDGE      PIC S9(9) BINARY VALUE 21.            780
COPIED        10 MQAT-NOTES-AGENT      PIC S9(9) BINARY VALUE 22.            781
COPIED        10 MQAT-USER             PIC S9(9) BINARY VALUE 25.            782
COPIED        10 MQAT-BROKER           PIC S9(9) BINARY VALUE 26.            783
COPIED        10 MQAT-JAVA             PIC S9(9) BINARY VALUE 28.            784

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    17
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQAT-DQM              PIC S9(9) BINARY VALUE 29.            785
COPIED        10 MQAT-CHANNEL-INITIATOR PIC S9(9) BINARY VALUE 30.           786
COPIED        10 MQAT-DEFAULT          PIC S9(9) BINARY VALUE 2.             787
COPIED        10 MQAT-USER-FIRST       PIC S9(9) BINARY VALUE 65536.         788
COPIED        10 MQAT-USER-LAST        PIC S9(9) BINARY VALUE 999999999.     789
```

Figure 22 MQGET Sample Compiled Listing (Page 10 of 35)

```
COPIED                                                                          790
COPIED      **   Group Identifier                                               791
COPIED        10 MQGI-NONE PIC X(24) VALUE LOW-VALUES.                         792
COPIED                                                                          793
COPIED      **   Message Flags                                                  794
COPIED        10 MQMF-SEGMENTATION-INHIBITED PIC S9(9) BINARY VALUE 0.         795
COPIED        10 MQMF-SEGMENTATION-ALLOWED   PIC S9(9) BINARY VALUE 1.         796
COPIED        10 MQMF-MSG-IN-GROUP           PIC S9(9) BINARY VALUE 8.         797
COPIED        10 MQMF-LAST-MSG-IN-GROUP      PIC S9(9) BINARY VALUE 16.        798
COPIED        10 MQMF-SEGMENT                PIC S9(9) BINARY VALUE 2.         799
COPIED        10 MQMF-LAST-SEGMENT           PIC S9(9) BINARY VALUE 4.         800
COPIED        10 MQMF-NONE                   PIC S9(9) BINARY VALUE 0.         801
COPIED                                                                          802
COPIED      **   Message Flags Masks                                            803
COPIED        10 MQMF-REJECT-UNSUP-MASK         PIC S9(9) BINARY VALUE 4095.  804
COPIED        10 MQMF-ACCEPT-UNSUP-MASK         PIC S9(9) BINARY VALUE        805
COPIED                                                          -1048576.       806
COPIED        10 MQMF-ACCEPT-UNSUP-IF-XMIT-MASK PIC S9(9) BINARY VALUE        807
COPIED                                                           1044480.       808
COPIED                                                                          809
COPIED      **   Original Length                                                810
COPIED        10 MQOL-UNDEFINED PIC S9(9) BINARY VALUE -1.                     811
COPIED                                                                          812
COPIED                                                                          813
COPIED      ******************************************************************  814
COPIED      **  Values Related to MQMDE Structure                           **  815
COPIED      ******************************************************************  816
COPIED                                                                          817
COPIED      **   Structure Identifier                                           818
COPIED        10 MQMDE-STRUC-ID PIC X(4) VALUE 'MDE '.                         819
COPIED                                                                          820
COPIED      **   Structure Version Number                                       821
COPIED        10 MQMDE-VERSION-2       PIC S9(9) BINARY VALUE 2.               822
COPIED        10 MQMDE-CURRENT-VERSION PIC S9(9) BINARY VALUE 2.               823
COPIED                                                                          824
COPIED      **   Structure Length                                               825
COPIED        10 MQMDE-LENGTH-2 PIC S9(9) BINARY VALUE 72.                     826
COPIED                                                                          827
COPIED      **   General Flags                                                  828
COPIED        10 MQMDEF-NONE PIC S9(9) BINARY VALUE 0.                         829
COPIED                                                                          830
COPIED                                                                          831
COPIED      ******************************************************************  832
COPIED      **  Values Related to MQOD Structure                            **  833

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    18
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED      ******************************************************************  834
COPIED                                                                          835
COPIED      **   Structure Identifier                                           836
COPIED        10 MQOD-STRUC-ID PIC X(4) VALUE 'OD  '.                          837
COPIED                                                                          838
COPIED      **   Structure Version Number                                       839
COPIED        10 MQOD-VERSION-1       PIC S9(9) BINARY VALUE 1.                840
COPIED        10 MQOD-VERSION-2       PIC S9(9) BINARY VALUE 2.                841
COPIED        10 MQOD-VERSION-3       PIC S9(9) BINARY VALUE 3.                842
COPIED        10 MQOD-CURRENT-VERSION PIC S9(9) BINARY VALUE 3.                843
COPIED                                                                          844
COPIED      **   Structure Length                                               845
COPIED        10 MQOD-CURRENT-LENGTH PIC S9(9) BINARY VALUE 336.               846
COPIED                                                                          847
COPIED      **   Object Types                                                   848
COPIED        10 MQOT-Q             PIC S9(9) BINARY VALUE 1.                  849
COPIED        10 MQOT-NAMELIST      PIC S9(9) BINARY VALUE 2.                  850
COPIED        10 MQOT-PROCESS       PIC S9(9) BINARY VALUE 3.                  851
COPIED        10 MQOT-STORAGE-CLASS PIC S9(9) BINARY VALUE 4.                  852
COPIED        10 MQOT-Q-MGR         PIC S9(9) BINARY VALUE 5.                  853
COPIED        10 MQOT-CHANNEL       PIC S9(9) BINARY VALUE 6.                  854
COPIED        10 MQOT-AUTH-INFO     PIC S9(9) BINARY VALUE 7.                  855
COPIED        10 MQOT-CF-STRUC      PIC S9(9) BINARY VALUE 10.                 856
COPIED        10 MQOT-RESERVED-1    PIC S9(9) BINARY VALUE 999.                857
COPIED                                                                          858
COPIED      **   Extended Object Types                                          859
COPIED        10 MQOT-ALL              PIC S9(9) BINARY VALUE 1001.            860
COPIED        10 MQOT-ALIAS-Q          PIC S9(9) BINARY VALUE 1002.            861
COPIED        10 MQOT-MODEL-Q          PIC S9(9) BINARY VALUE 1003.            862
COPIED        10 MQOT-LOCAL-Q          PIC S9(9) BINARY VALUE 1004.            863
COPIED        10 MQOT-REMOTE-Q         PIC S9(9) BINARY VALUE 1005.            864
COPIED        10 MQOT-SENDER-CHANNEL   PIC S9(9) BINARY VALUE 1007.            865
COPIED        10 MQOT-SERVER-CHANNEL   PIC S9(9) BINARY VALUE 1008.            866
COPIED        10 MQOT-REQUESTER-CHANNEL PIC S9(9) BINARY VALUE 1009.           867
COPIED        10 MQOT-RECEIVER-CHANNEL  PIC S9(9) BINARY VALUE 1010.           868
COPIED        10 MQOT-CURRENT-CHANNEL   PIC S9(9) BINARY VALUE 1011.           869
COPIED        10 MQOT-SAVED-CHANNEL     PIC S9(9) BINARY VALUE 1012.           870
COPIED        10 MQOT-SVRCONN-CHANNEL   PIC S9(9) BINARY VALUE 1013.           871
```

Figure 22 MQGET Sample Compiled Listing (Page 11 of 35)

```
COPIED        10 MQOT-CLNTCONN-CHANNEL  PIC S9(9) BINARY VALUE 1014.                    872
COPIED                                                                                 873
COPIED      **   Security Identifier                                                   874
COPIED        10 MQSID-NONE PIC X(40) VALUE LOW-VALUES.                                875
COPIED                                                                                 876
COPIED      **   Security Identifier Type                                              877
COPIED        10 MQSIDT-NONE            PIC X VALUE X'00'.                           878
COPIED        10 MQSIDT-NT-SECURITY-ID PIC X VALUE X'01'.                            879
COPIED                                                                                 880
COPIED                                                                                 881
COPIED      ******************************************************************       882

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    19
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED      **  Values Related to MQPMO Structure                          **          883
COPIED      ******************************************************************       884
COPIED                                                                                 885
COPIED      **   Structure Identifier                                                  886
COPIED        10 MQPMO-STRUC-ID PIC X(4) VALUE 'PMO '.                                 887
COPIED                                                                                 888
COPIED      **   Structure Version Number                                              889
COPIED        10 MQPMO-VERSION-1       PIC S9(9) BINARY VALUE 1.                     890
COPIED        10 MQPMO-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.                      891
COPIED                                                                                 892
COPIED      **   Structure Length                                                      893
COPIED        10 MQPMO-CURRENT-LENGTH PIC S9(9) BINARY VALUE 128.                      894
COPIED                                                                                 895
COPIED      **   Put-Message Options                                                   896
COPIED        10 MQPMO-SYNCPOINT               PIC S9(9) BINARY VALUE 2.           897
COPIED        10 MQPMO-NO-SYNCPOINT            PIC S9(9) BINARY VALUE 4.           898
COPIED        10 MQPMO-LOGICAL-ORDER           PIC S9(9) BINARY VALUE 32768.       899
COPIED        10 MQPMO-NO-CONTEXT              PIC S9(9) BINARY VALUE 16384.       900
COPIED        10 MQPMO-DEFAULT-CONTEXT         PIC S9(9) BINARY VALUE 32.          901
COPIED        10 MQPMO-PASS-IDENTITY-CONTEXT   PIC S9(9) BINARY VALUE 256.         902
COPIED        10 MQPMO-PASS-ALL-CONTEXT        PIC S9(9) BINARY VALUE 512.         903
COPIED        10 MQPMO-SET-IDENTITY-CONTEXT    PIC S9(9) BINARY VALUE 1024.        904
COPIED        10 MQPMO-SET-ALL-CONTEXT         PIC S9(9) BINARY VALUE 2048.        905
COPIED        10 MQPMO-ALTERNATE-USER-AUTHORITY PIC S9(9) BINARY VALUE 4096.       906
COPIED        10 MQPMO-FAIL-IF-QUIESCING       PIC S9(9) BINARY VALUE 8192.        907
COPIED        10 MQPMO-NONE                    PIC S9(9) BINARY VALUE 0.           908
COPIED                                                                                 909
COPIED                                                                                 910
COPIED      ******************************************************************       911
COPIED      **  Values Related to MQRFH Structure                          **          912
COPIED      ******************************************************************       913
COPIED                                                                                 914
COPIED      **   Structure Identifier                                                  915
COPIED        10 MQRFH-STRUC-ID PIC X(4) VALUE 'RFH '.                                 916
COPIED                                                                                 917
COPIED      **   Structure Version Number                                              918
COPIED        10 MQRFH-VERSION-1 PIC S9(9) BINARY VALUE 1.                             919
COPIED        10 MQRFH-VERSION-2 PIC S9(9) BINARY VALUE 2.                             920
COPIED                                                                                 921
COPIED      **   Structure Length                                                      922
COPIED        10 MQRFH-STRUC-LENGTH-FIXED   PIC S9(9) BINARY VALUE 32.                923
COPIED        10 MQRFH-STRUC-LENGTH-FIXED-2 PIC S9(9) BINARY VALUE 36.                 924
COPIED                                                                                 925
COPIED      **   Flags                                                                 926
COPIED        10 MQRFH-NONE PIC S9(9) BINARY VALUE 0.                                  927
COPIED                                                                                 928
COPIED      **   Names for Name/Value String                                           929
COPIED        10 MQNVS-APPL-TYPE PIC X(12) VALUE 'OPT_APP_GRP '.                       930
COPIED        10 MQNVS-MSG-TYPE  PIC X(13) VALUE 'OPT_MSG_TYPE '.                      931

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    20
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED                                                                                 932
COPIED                                                                                 933
COPIED      ******************************************************************       934
COPIED      **  Values Related to MQRMH Structure                          **          935
COPIED      ******************************************************************       936
COPIED                                                                                 937
COPIED      **   Structure Identifier                                                  938
COPIED        10 MQRMH-STRUC-ID PIC X(4) VALUE 'RMH '.                                 939
COPIED                                                                                 940
COPIED      **   Structure Version Number                                              941
COPIED        10 MQRMH-VERSION-1       PIC S9(9) BINARY VALUE 1.                     942
COPIED        10 MQRMH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.                      943
COPIED                                                                                 944
COPIED      **   Flags                                                                 945
COPIED        10 MQRMHF-LAST     PIC S9(9) BINARY VALUE 1.                           946
COPIED        10 MQRMHF-NOT-LAST PIC S9(9) BINARY VALUE 0.                           947
COPIED                                                                                 948
COPIED      **   Object Instance Identifier                                            949
COPIED        10 MQOII-NONE PIC X(24) VALUE LOW-VALUES.                                950
```

Figure 22 MQGET Sample Compiled Listing (Page 12 of 35)

```
COPIED                                                                          951
COPIED                                                                          952
COPIED      ****************************************************************** 953
COPIED      **  Values Related to MQIM Structure                            ** 954
COPIED      ****************************************************************** 955
COPIED                                                                          956
COPIED      **   Structure Identifier                                           957
COPIED        10 MQIM-STRUC-ID PIC X(4) VALUE 'IM  '.                          958
COPIED                                                                          959
COPIED      **   Structure Version Number                                       960
COPIED        10 MQIM-VERSION-1       PIC S9(9) BINARY VALUE 1.                961
COPIED        10 MQIM-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.                962
COPIED                                                                          963
COPIED                                                                          964
COPIED      ****************************************************************** 965
COPIED      **  Values Related to MQIMC2 Structure                          ** 966
COPIED      ****************************************************************** 967
COPIED                                                                          968
COPIED      **   Structure Identifier                                           969
COPIED        10 MQIMC-STRUC-ID PIC X(4) VALUE 'IMC '.                         970
COPIED                                                                          971
COPIED      **   Structure Version Number                                       972
COPIED        10 MQIMC-VERSION-1       PIC X(4) VALUE '   1'.                 973
COPIED        10 MQIMC-VERSION-2       PIC X(4) VALUE '   2'.                 974
COPIED        10 MQIMC-CURRENT-VERSION PIC X(4) VALUE '   2'.                 975
COPIED                                                                          976
COPIED                                                                          977
COPIED      ****************************************************************** 978
COPIED      **  Values Related to MQWIH Structure                           ** 979
COPIED      ****************************************************************** 980

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    21
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED                                                                          981
COPIED      **   Structure Identifier                                           982
COPIED        10 MQWIH-STRUC-ID PIC X(4) VALUE 'WIH '.                         983
COPIED                                                                          984
COPIED      **   Structure Version Number                                       985
COPIED        10 MQWIH-VERSION-1       PIC S9(9) BINARY VALUE 1.               986
COPIED        10 MQWIH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.               987
COPIED                                                                          988
COPIED      **   Structure Length                                               989
COPIED        10 MQWIH-LENGTH-1       PIC S9(9) BINARY VALUE 120.              990
COPIED        10 MQWIH-CURRENT-LENGTH PIC S9(9) BINARY VALUE 120.              991
COPIED                                                                          992
COPIED      **   Flags                                                          993
COPIED        10 MQWIH-NONE PIC S9(9) BINARY VALUE 0.                          994
COPIED                                                                          995
COPIED                                                                          996
COPIED      ****************************************************************** 997
COPIED      **  Values Related to MQXQH Structure                           ** 998
COPIED      ****************************************************************** 999
COPIED                                                                         1000
COPIED      **   Structure Identifier                                          1001
COPIED        10 MQXQH-STRUC-ID PIC X(4) VALUE 'XQH '.                        1002
COPIED                                                                         1003
COPIED      **   Structure Version Number                                      1004
COPIED        10 MQXQH-VERSION-1       PIC S9(9) BINARY VALUE 1.              1005
COPIED        10 MQXQH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.              1006
COPIED                                                                         1007
COPIED                                                                         1008
COPIED      ****************************************************************** 1009
COPIED      **  Values Related to MQCLOSE Call                              ** 1010
COPIED      ****************************************************************** 1011
COPIED                                                                         1012
COPIED      **   Close Options                                                 1013
COPIED        10 MQCO-NONE         PIC S9(9) BINARY VALUE 0.                 1014
COPIED        10 MQCO-DELETE       PIC S9(9) BINARY VALUE 1.                 1015
COPIED        10 MQCO-DELETE-PURGE PIC S9(9) BINARY VALUE 2.                 1016
COPIED                                                                         1017
COPIED                                                                         1018
COPIED      ****************************************************************** 1019
COPIED      **  Values Related to MQINQ Call                                ** 1020
COPIED      ****************************************************************** 1021
COPIED                                                                         1022
COPIED      **   Byte-Attribute Selectors                                      1023
COPIED        10 MQBA-FIRST PIC S9(9) BINARY VALUE 6001.                      1024
COPIED        10 MQBA-LAST  PIC S9(9) BINARY VALUE 8000.                      1025
COPIED                                                                         1026
COPIED      **   Character-Attribute Selectors                                 1027
COPIED        10 MQCA-ALTERATION-DATE       PIC S9(9) BINARY VALUE 2027.     1028
COPIED        10 MQCA-ALTERATION-TIME       PIC S9(9) BINARY VALUE 2028.     1029

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    22
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
```

Figure 22 MQGET Sample Compiled Listing (Page 13 of 35)

```
COPIED        10 MQCA-APPL-ID                PIC S9(9) BINARY VALUE 2001.                    1030
COPIED        10 MQCA-AUTH-INFO-CONN-NAME    PIC S9(9) BINARY VALUE 2053.                    1031
COPIED        10 MQCA-AUTH-INFO-DESC         PIC S9(9) BINARY VALUE 2046.                    1032
COPIED        10 MQCA-AUTH-INFO-NAME         PIC S9(9) BINARY VALUE 2045.                    1033
COPIED        10 MQCA-BACKOUT-REQ-Q-NAME     PIC S9(9) BINARY VALUE 2019.                    1034
COPIED        10 MQCA-BASE-Q-NAME            PIC S9(9) BINARY VALUE 2002.                    1035
COPIED        10 MQCA-CF-STRUC-DESC          PIC S9(9) BINARY VALUE 2052.                    1036
COPIED        10 MQCA-CF-STRUC-NAME          PIC S9(9) BINARY VALUE 2039.                    1037
COPIED        10 MQCA-CHANNEL-AUTO-DEF-EXIT  PIC S9(9) BINARY VALUE 2026.                    1038
COPIED        10 MQCA-CLUSTER-DATE           PIC S9(9) BINARY VALUE 2037.                    1039
COPIED        10 MQCA-CLUSTER-NAME           PIC S9(9) BINARY VALUE 2029.                    1040
COPIED        10 MQCA-CLUSTER-NAMELIST       PIC S9(9) BINARY VALUE 2030.                    1041
COPIED        10 MQCA-CLUSTER-Q-MGR-NAME     PIC S9(9) BINARY VALUE 2031.                    1042
COPIED        10 MQCA-CLUSTER-TIME           PIC S9(9) BINARY VALUE 2038.                    1043
COPIED        10 MQCA-CLUSTER-WORKLOAD-DATA  PIC S9(9) BINARY VALUE 2034.                    1044
COPIED        10 MQCA-CLUSTER-WORKLOAD-EXIT  PIC S9(9) BINARY VALUE 2033.                    1045
COPIED        10 MQCA-COMMAND-INPUT-Q-NAME   PIC S9(9) BINARY VALUE 2003.                    1046
COPIED        10 MQCA-CREATION-DATE          PIC S9(9) BINARY VALUE 2004.                    1047
COPIED        10 MQCA-CREATION-TIME          PIC S9(9) BINARY VALUE 2005.                    1048
COPIED        10 MQCA-DEAD-LETTER-Q-NAME     PIC S9(9) BINARY VALUE 2006.                    1049
COPIED        10 MQCA-DEF-XMIT-Q-NAME        PIC S9(9) BINARY VALUE 2025.                    1050
COPIED        10 MQCA-ENV-DATA               PIC S9(9) BINARY VALUE 2007.                    1051
COPIED        10 MQCA-FIRST                  PIC S9(9) BINARY VALUE 2001.                    1052
COPIED        10 MQCA-IGQ-USER-ID            PIC S9(9) BINARY VALUE 2041.                    1053
COPIED        10 MQCA-INITIATION-Q-NAME      PIC S9(9) BINARY VALUE 2008.                    1054
COPIED        10 MQCA-LAST                   PIC S9(9) BINARY VALUE 4000.                    1055
COPIED        10 MQCA-LAST-USED              PIC S9(9) BINARY VALUE 2053.                    1056
COPIED        10 MQCA-LDAP-PASSWORD          PIC S9(9) BINARY VALUE 2048.                    1057
COPIED        10 MQCA-LDAP-USER-NAME         PIC S9(9) BINARY VALUE 2047.                    1058
COPIED        10 MQCA-NAMELIST-DESC          PIC S9(9) BINARY VALUE 2009.                    1059
COPIED        10 MQCA-NAMELIST-NAME          PIC S9(9) BINARY VALUE 2010.                    1060
COPIED        10 MQCA-NAMES                  PIC S9(9) BINARY VALUE 2020.                    1061
COPIED        10 MQCA-PROCESS-DESC           PIC S9(9) BINARY VALUE 2011.                    1062
COPIED        10 MQCA-PROCESS-NAME           PIC S9(9) BINARY VALUE 2012.                    1063
COPIED        10 MQCA-Q-DESC                 PIC S9(9) BINARY VALUE 2013.                    1064
COPIED        10 MQCA-Q-MGR-DESC             PIC S9(9) BINARY VALUE 2014.                    1065
COPIED        10 MQCA-Q-MGR-IDENTIFIER       PIC S9(9) BINARY VALUE 2032.                    1066
COPIED        10 MQCA-Q-MGR-NAME             PIC S9(9) BINARY VALUE 2015.                    1067
COPIED        10 MQCA-Q-NAME                 PIC S9(9) BINARY VALUE 2016.                    1068
COPIED        10 MQCA-QSG-NAME               PIC S9(9) BINARY VALUE 2040.                    1069
COPIED        10 MQCA-REMOTE-Q-MGR-NAME      PIC S9(9) BINARY VALUE 2017.                    1070
COPIED        10 MQCA-REMOTE-Q-NAME          PIC S9(9) BINARY VALUE 2018.                    1071
COPIED        10 MQCA-REPOSITORY-NAME        PIC S9(9) BINARY VALUE 2035.                    1072
COPIED        10 MQCA-REPOSITORY-NAMELIST    PIC S9(9) BINARY VALUE 2036.                    1073
COPIED        10 MQCA-SSL-CRL-NAMELIST       PIC S9(9) BINARY VALUE 2050.                    1074
COPIED        10 MQCA-SSL-CRYPTO-HARDWARE    PIC S9(9) BINARY VALUE 2051.                    1075
COPIED        10 MQCA-SSL-KEY-REPOSITORY     PIC S9(9) BINARY VALUE 2049.                    1076
COPIED        10 MQCA-STORAGE-CLASS          PIC S9(9) BINARY VALUE 2022.                    1077
COPIED        10 MQCA-STORAGE-CLASS-DESC     PIC S9(9) BINARY VALUE 2042.                    1078

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    23
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQCA-TRIGGER-DATA           PIC S9(9) BINARY VALUE 2023.                    1079
COPIED        10 MQCA-USER-DATA              PIC S9(9) BINARY VALUE 2021.                    1080
COPIED        10 MQCA-USER-LIST              PIC S9(9) BINARY VALUE 4000.                    1081
COPIED        10 MQCA-XCF-GROUP-NAME         PIC S9(9) BINARY VALUE 2043.                    1082
COPIED        10 MQCA-XCF-MEMBER-NAME        PIC S9(9) BINARY VALUE 2044.                    1083
COPIED        10 MQCA-XMIT-Q-NAME            PIC S9(9) BINARY VALUE 2024.                    1084
COPIED                                                                                       1085
COPIED      **   Integer-Attribute Selectors                                                 1086
COPIED        10 MQIA-APPL-TYPE                PIC S9(9) BINARY VALUE 1.                   1087
COPIED        10 MQIA-ARCHIVE                  PIC S9(9) BINARY VALUE 60.                  1088
COPIED        10 MQIA-AUTH-INFO-TYPE           PIC S9(9) BINARY VALUE 66.                  1089
COPIED        10 MQIA-AUTHORITY-EVENT          PIC S9(9) BINARY VALUE 47.                  1090
COPIED        10 MQIA-BACKOUT-THRESHOLD        PIC S9(9) BINARY VALUE 22.                  1091
COPIED        10 MQIA-CF-LEVEL                 PIC S9(9) BINARY VALUE 70.                  1092
COPIED        10 MQIA-CF-RECOVER               PIC S9(9) BINARY VALUE 71.                  1093
COPIED        10 MQIA-CHANNEL-AUTO-DEF         PIC S9(9) BINARY VALUE 55.                  1094
COPIED        10 MQIA-CHANNEL-AUTO-DEF-EVENT   PIC S9(9) BINARY VALUE 56.                  1095
COPIED        10 MQIA-CLUSTER-Q-TYPE           PIC S9(9) BINARY VALUE 59.                  1096
COPIED        10 MQIA-CLUSTER-WORKLOAD-LENGTH  PIC S9(9) BINARY VALUE 58.                  1097
COPIED        10 MQIA-CODED-CHAR-SET-ID        PIC S9(9) BINARY VALUE 2.                   1098
COPIED        10 MQIA-COMMAND-LEVEL            PIC S9(9) BINARY VALUE 31.                  1099
COPIED        10 MQIA-CONFIGURATION-EVENT      PIC S9(9) BINARY VALUE 51.                  1100
COPIED        10 MQIA-CPI-LEVEL                PIC S9(9) BINARY VALUE 27.                  1101
COPIED        10 MQIA-CURRENT-Q-DEPTH          PIC S9(9) BINARY VALUE 3.                   1102
COPIED        10 MQIA-DEF-BIND                 PIC S9(9) BINARY VALUE 61.                  1103
COPIED        10 MQIA-DEF-INPUT-OPEN-OPTION    PIC S9(9) BINARY VALUE 4.                   1104
COPIED        10 MQIA-DEF-PERSISTENCE          PIC S9(9) BINARY VALUE 5.                   1105
COPIED        10 MQIA-DEF-PRIORITY             PIC S9(9) BINARY VALUE 6.                   1106
COPIED        10 MQIA-DEFINITION-TYPE          PIC S9(9) BINARY VALUE 7.                   1107
COPIED        10 MQIA-DIST-LISTS               PIC S9(9) BINARY VALUE 34.                  1108
COPIED        10 MQIA-EXPIRY-INTERVAL          PIC S9(9) BINARY VALUE 39.                  1109
COPIED        10 MQIA-FIRST                    PIC S9(9) BINARY VALUE 1.                   1110
COPIED        10 MQIA-HARDEN-GET-BACKOUT       PIC S9(9) BINARY VALUE 8.                   1111
```

Figure 22 MQGET Sample Compiled Listing (Page 14 of 35)

```
COPIED        10 MQIA-HIGH-Q-DEPTH            PIC S9(9) BINARY VALUE 36.                          1112
COPIED        10 MQIA-IGQ-PUT-AUTHORITY       PIC S9(9) BINARY VALUE 65.                          1113
COPIED        10 MQIA-INDEX-TYPE              PIC S9(9) BINARY VALUE 57.                          1114
COPIED        10 MQIA-INHIBIT-EVENT           PIC S9(9) BINARY VALUE 48.                          1115
COPIED        10 MQIA-INHIBIT-GET             PIC S9(9) BINARY VALUE 9.                           1116
COPIED        10 MQIA-INHIBIT-PUT             PIC S9(9) BINARY VALUE 10.                          1117
COPIED        10 MQIA-INTRA-GROUP-QUEUING     PIC S9(9) BINARY VALUE 64.                          1118
COPIED        10 MQIA-LAST                    PIC S9(9) BINARY VALUE 2000.                        1119
COPIED        10 MQIA-LAST-USED               PIC S9(9) BINARY VALUE 72.                          1120
COPIED        10 MQIA-LOCAL-EVENT             PIC S9(9) BINARY VALUE 49.                          1121
COPIED        10 MQIA-MAX-HANDLES             PIC S9(9) BINARY VALUE 11.                          1122
COPIED        10 MQIA-MAX-MSG-LENGTH          PIC S9(9) BINARY VALUE 13.                          1123
COPIED        10 MQIA-MAX-PRIORITY            PIC S9(9) BINARY VALUE 14.                          1124
COPIED        10 MQIA-MAX-Q-DEPTH             PIC S9(9) BINARY VALUE 15.                          1125
COPIED        10 MQIA-MAX-UNCOMMITTED-MSGS    PIC S9(9) BINARY VALUE 33.                          1126
COPIED        10 MQIA-MSG-DELIVERY-SEQUENCE   PIC S9(9) BINARY VALUE 16.                          1127

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    24
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQIA-MSG-DEQ-COUNT           PIC S9(9) BINARY VALUE 38.                          1128
COPIED        10 MQIA-MSG-ENQ-COUNT           PIC S9(9) BINARY VALUE 37.                          1129
COPIED        10 MQIA-NAME-COUNT              PIC S9(9) BINARY VALUE 19.                          1130
COPIED        10 MQIA-NAMELIST-TYPE           PIC S9(9) BINARY VALUE 72.                          1131
COPIED        10 MQIA-OPEN-INPUT-COUNT        PIC S9(9) BINARY VALUE 17.                          1132
COPIED        10 MQIA-OPEN-OUTPUT-COUNT       PIC S9(9) BINARY VALUE 18.                          1133
COPIED        10 MQIA-PAGESET-ID              PIC S9(9) BINARY VALUE 62.                          1134
COPIED        10 MQIA-PERFORMANCE-EVENT       PIC S9(9) BINARY VALUE 53.                          1135
COPIED        10 MQIA-PLATFORM                PIC S9(9) BINARY VALUE 32.                          1136
COPIED        10 MQIA-Q-DEPTH-HIGH-EVENT      PIC S9(9) BINARY VALUE 43.                          1137
COPIED        10 MQIA-Q-DEPTH-HIGH-LIMIT      PIC S9(9) BINARY VALUE 40.                          1138
COPIED        10 MQIA-Q-DEPTH-LOW-EVENT       PIC S9(9) BINARY VALUE 44.                          1139
COPIED        10 MQIA-Q-DEPTH-LOW-LIMIT       PIC S9(9) BINARY VALUE 41.                          1140
COPIED        10 MQIA-Q-DEPTH-MAX-EVENT       PIC S9(9) BINARY VALUE 42.                          1141
COPIED        10 MQIA-Q-SERVICE-INTERVAL      PIC S9(9) BINARY VALUE 54.                          1142
COPIED        10 MQIA-Q-SERVICE-INTERVAL-EVENT PIC S9(9) BINARY VALUE 46.                         1143
COPIED        10 MQIA-Q-TYPE                  PIC S9(9) BINARY VALUE 20.                          1144
COPIED        10 MQIA-QSG-DISP                PIC S9(9) BINARY VALUE 63.                          1145
COPIED        10 MQIA-REMOTE-EVENT            PIC S9(9) BINARY VALUE 50.                          1146
COPIED        10 MQIA-RETENTION-INTERVAL      PIC S9(9) BINARY VALUE 21.                          1147
COPIED        10 MQIA-SCOPE                   PIC S9(9) BINARY VALUE 45.                          1148
COPIED        10 MQIA-SHAREABILITY            PIC S9(9) BINARY VALUE 23.                          1149
COPIED        10 MQIA-SSL-TASKS               PIC S9(9) BINARY VALUE 69.                          1150
COPIED        10 MQIA-START-STOP-EVENT        PIC S9(9) BINARY VALUE 52.                          1151
COPIED        10 MQIA-SYNCPOINT               PIC S9(9) BINARY VALUE 30.                          1152
COPIED        10 MQIA-TIME-SINCE-RESET        PIC S9(9) BINARY VALUE 35.                          1153
COPIED        10 MQIA-TRIGGER-CONTROL         PIC S9(9) BINARY VALUE 24.                          1154
COPIED        10 MQIA-TRIGGER-DEPTH           PIC S9(9) BINARY VALUE 29.                          1155
COPIED        10 MQIA-TRIGGER-INTERVAL        PIC S9(9) BINARY VALUE 25.                          1156
COPIED        10 MQIA-TRIGGER-MSG-PRIORITY    PIC S9(9) BINARY VALUE 26.                          1157
COPIED        10 MQIA-TRIGGER-TYPE            PIC S9(9) BINARY VALUE 28.                          1158
COPIED        10 MQIA-USAGE                   PIC S9(9) BINARY VALUE 12.                          1159
COPIED        10 MQIA-USER-LIST               PIC S9(9) BINARY VALUE 2000.                        1160
COPIED                                                                                            1161
COPIED      **   Integer Attribute Value Denoting "Not Applicable"                                 1162
COPIED        10 MQIAV-NOT-APPLICABLE PIC S9(9) BINARY VALUE -1.                                  1163
COPIED        10 MQIAV-UNDEFINED      PIC S9(9) BINARY VALUE -2.                                  1164
COPIED                                                                                            1165
COPIED                                                                                            1166
COPIED      ************************************************************************            1167
COPIED      **  Values Related to MQOPEN Call                                       **            1168
COPIED      ************************************************************************            1169
COPIED                                                                                            1170
COPIED      **   Open Options                                                                     1171
COPIED        10 MQOO-INPUT-AS-Q-DEF          PIC S9(9) BINARY VALUE 1.                           1172
COPIED        10 MQOO-INPUT-SHARED            PIC S9(9) BINARY VALUE 2.                           1173
COPIED        10 MQOO-INPUT-EXCLUSIVE         PIC S9(9) BINARY VALUE 4.                           1174
COPIED        10 MQOO-BROWSE                  PIC S9(9) BINARY VALUE 8.                           1175
COPIED        10 MQOO-OUTPUT                  PIC S9(9) BINARY VALUE 16.                          1176

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    25
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQOO-INQUIRE                 PIC S9(9) BINARY VALUE 32.                          1177
COPIED        10 MQOO-SET                     PIC S9(9) BINARY VALUE 64.                          1178
COPIED        10 MQOO-BIND-ON-OPEN            PIC S9(9) BINARY VALUE 16384.                       1179
COPIED        10 MQOO-BIND-NOT-FIXED          PIC S9(9) BINARY VALUE 32768.                       1180
COPIED        10 MQOO-BIND-AS-Q-DEF           PIC S9(9) BINARY VALUE 0.                           1181
COPIED        10 MQOO-SAVE-ALL-CONTEXT        PIC S9(9) BINARY VALUE 128.                         1182
COPIED        10 MQOO-PASS-IDENTITY-CONTEXT   PIC S9(9) BINARY VALUE 256.                         1183
COPIED        10 MQOO-PASS-ALL-CONTEXT        PIC S9(9) BINARY VALUE 512.                         1184
COPIED        10 MQOO-SET-IDENTITY-CONTEXT    PIC S9(9) BINARY VALUE 1024.                        1185
COPIED        10 MQOO-SET-ALL-CONTEXT         PIC S9(9) BINARY VALUE 2048.                        1186
COPIED        10 MQOO-ALTERNATE-USER-AUTHORITY PIC S9(9) BINARY VALUE 4096.                       1187
COPIED        10 MQOO-FAIL-IF-QUIESCING       PIC S9(9) BINARY VALUE 8192.                        1188
COPIED                                                                                            1189
COPIED                                                                                            1190
```

Figure 22 MQGET Sample Compiled Listing (Page 15 of 35)

```
COPIED      ****************************************************************          1191
COPIED      ** Values Related to All Calls                                **          1192
COPIED      ****************************************************************          1193
COPIED                                                                                1194
COPIED      **   Connection Handle                                                    1195
COPIED        10 MQHC-DEF-HCONN PIC S9(9) BINARY VALUE 0.                             1196
COPIED                                                                                1197
COPIED      **   String Lengths                                                       1198
COPIED        10 MQ-ABEND-CODE-LENGTH          PIC S9(9) BINARY VALUE 4.              1199
COPIED        10 MQ-ACCOUNTING-TOKEN-LENGTH    PIC S9(9) BINARY VALUE 32.             1200
COPIED        10 MQ-APPL-IDENTITY-DATA-LENGTH  PIC S9(9) BINARY VALUE 32.             1201
COPIED        10 MQ-APPL-NAME-LENGTH           PIC S9(9) BINARY VALUE 28.             1202
COPIED        10 MQ-APPL-ORIGIN-DATA-LENGTH    PIC S9(9) BINARY VALUE 4.              1203
COPIED        10 MQ-APPL-TAG-LENGTH            PIC S9(9) BINARY VALUE 28.             1204
COPIED        10 MQ-ATTENTION-ID-LENGTH        PIC S9(9) BINARY VALUE 4.              1205
COPIED        10 MQ-AUTH-INFO-CONN-NAME-LENGTH PIC S9(9) BINARY VALUE 264.            1206
COPIED        10 MQ-AUTH-INFO-DESC-LENGTH      PIC S9(9) BINARY VALUE 64.             1207
COPIED        10 MQ-AUTH-INFO-NAME-LENGTH      PIC S9(9) BINARY VALUE 48.             1208
COPIED        10 MQ-AUTHENTICATOR-LENGTH       PIC S9(9) BINARY VALUE 8.              1209
COPIED        10 MQ-BRIDGE-NAME-LENGTH         PIC S9(9) BINARY VALUE 24.             1210
COPIED        10 MQ-CANCEL-CODE-LENGTH         PIC S9(9) BINARY VALUE 4.              1211
COPIED        10 MQ-CF-STRUC-DESC-LENGTH       PIC S9(9) BINARY VALUE 64.             1212
COPIED        10 MQ-CF-STRUC-NAME-LENGTH       PIC S9(9) BINARY VALUE 12.             1213
COPIED        10 MQ-CHANNEL-DATE-LENGTH        PIC S9(9) BINARY VALUE 12.             1214
COPIED        10 MQ-CHANNEL-DESC-LENGTH        PIC S9(9) BINARY VALUE 64.             1215
COPIED        10 MQ-CHANNEL-NAME-LENGTH        PIC S9(9) BINARY VALUE 20.             1216
COPIED        10 MQ-CHANNEL-TIME-LENGTH        PIC S9(9) BINARY VALUE 8.              1217
COPIED        10 MQ-CLUSTER-NAME-LENGTH        PIC S9(9) BINARY VALUE 48.             1218
COPIED        10 MQ-CONN-NAME-LENGTH           PIC S9(9) BINARY VALUE 264.            1219
COPIED        10 MQ-CONN-TAG-LENGTH            PIC S9(9) BINARY VALUE 128.            1220
COPIED        10 MQ-CORREL-ID-LENGTH           PIC S9(9) BINARY VALUE 24.             1221
COPIED        10 MQ-CREATION-DATE-LENGTH       PIC S9(9) BINARY VALUE 12.             1222
COPIED        10 MQ-CREATION-TIME-LENGTH       PIC S9(9) BINARY VALUE 8.              1223
COPIED        10 MQ-DATE-LENGTH                PIC S9(9) BINARY VALUE 12.             1224
COPIED        10 MQ-DISTINGUISHED-NAME-LENGTH  PIC S9(9) BINARY VALUE 1024.           1225

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    26
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQ-EXIT-DATA-LENGTH           PIC S9(9) BINARY VALUE 32.             1226
COPIED        10 MQ-EXIT-NAME-LENGTH           PIC S9(9) BINARY VALUE 8.              1227
COPIED        10 MQ-EXIT-USER-AREA-LENGTH      PIC S9(9) BINARY VALUE 16.             1228
COPIED        10 MQ-FACILITY-LENGTH            PIC S9(9) BINARY VALUE 8.              1229
COPIED        10 MQ-FACILITY-LIKE-LENGTH       PIC S9(9) BINARY VALUE 4.              1230
COPIED        10 MQ-FORMAT-LENGTH              PIC S9(9) BINARY VALUE 8.              1231
COPIED        10 MQ-FUNCTION-LENGTH            PIC S9(9) BINARY VALUE 4.              1232
COPIED        10 MQ-GROUP-ID-LENGTH            PIC S9(9) BINARY VALUE 24.             1233
COPIED        10 MQ-LDAP-PASSWORD-LENGTH       PIC S9(9) BINARY VALUE 32.             1234
COPIED        10 MQ-LOCAL-ADDRESS-LENGTH       PIC S9(9) BINARY VALUE 48.             1235
COPIED        10 MQ-LTERM-OVERRIDE-LENGTH      PIC S9(9) BINARY VALUE 8.              1236
COPIED        10 MQ-LUWID-LENGTH               PIC S9(9) BINARY VALUE 16.             1237
COPIED        10 MQ-MAX-EXIT-NAME-LENGTH       PIC S9(9) BINARY VALUE 128.            1238
COPIED        10 MQ-MAX-MCA-USER-ID-LENGTH     PIC S9(9) BINARY VALUE 64.             1239
COPIED        10 MQ-MAX-USER-ID-LENGTH         PIC S9(9) BINARY VALUE 64.             1240
COPIED        10 MQ-MCA-JOB-NAME-LENGTH        PIC S9(9) BINARY VALUE 28.             1241
COPIED        10 MQ-MCA-NAME-LENGTH            PIC S9(9) BINARY VALUE 20.             1242
COPIED        10 MQ-MCA-USER-DATA-LENGTH       PIC S9(9) BINARY VALUE 32.             1243
COPIED        10 MQ-MCA-USER-ID-LENGTH         PIC S9(9) BINARY VALUE 12.             1244
COPIED        10 MQ-MFS-MAP-NAME-LENGTH        PIC S9(9) BINARY VALUE 8.              1245
COPIED        10 MQ-MODE-NAME-LENGTH           PIC S9(9) BINARY VALUE 8.              1246
COPIED        10 MQ-MSG-HEADER-LENGTH          PIC S9(9) BINARY VALUE 4000.           1247
COPIED        10 MQ-MSG-ID-LENGTH              PIC S9(9) BINARY VALUE 24.             1248
COPIED        10 MQ-MSG-TOKEN-LENGTH           PIC S9(9) BINARY VALUE 16.             1249
COPIED        10 MQ-NAMELIST-DESC-LENGTH       PIC S9(9) BINARY VALUE 64.             1250
COPIED        10 MQ-NAMELIST-NAME-LENGTH       PIC S9(9) BINARY VALUE 48.             1251
COPIED        10 MQ-OBJECT-INSTANCE-ID-LENGTH  PIC S9(9) BINARY VALUE 24.             1252
COPIED        10 MQ-OBJECT-NAME-LENGTH         PIC S9(9) BINARY VALUE 48.             1253
COPIED        10 MQ-PASSWORD-LENGTH            PIC S9(9) BINARY VALUE 12.             1254
COPIED        10 MQ-PROCESS-APPL-ID-LENGTH     PIC S9(9) BINARY VALUE 256.            1255
COPIED        10 MQ-PROCESS-DESC-LENGTH        PIC S9(9) BINARY VALUE 64.             1256
COPIED        10 MQ-PROCESS-ENV-DATA-LENGTH    PIC S9(9) BINARY VALUE 128.            1257
COPIED        10 MQ-PROCESS-NAME-LENGTH        PIC S9(9) BINARY VALUE 48.             1258
COPIED        10 MQ-PROCESS-USER-DATA-LENGTH   PIC S9(9) BINARY VALUE 128.            1259
COPIED        10 MQ-PROGRAM-NAME-LENGTH        PIC S9(9) BINARY VALUE 20.             1260
COPIED        10 MQ-PUT-APPL-NAME-LENGTH       PIC S9(9) BINARY VALUE 28.             1261
COPIED        10 MQ-PUT-DATE-LENGTH            PIC S9(9) BINARY VALUE 8.              1262
COPIED        10 MQ-PUT-TIME-LENGTH            PIC S9(9) BINARY VALUE 8.              1263
COPIED        10 MQ-Q-DESC-LENGTH              PIC S9(9) BINARY VALUE 64.             1264
COPIED        10 MQ-Q-MGR-DESC-LENGTH          PIC S9(9) BINARY VALUE 64.             1265
COPIED        10 MQ-Q-MGR-IDENTIFIER-LENGTH    PIC S9(9) BINARY VALUE 48.             1266
COPIED        10 MQ-Q-MGR-NAME-LENGTH          PIC S9(9) BINARY VALUE 48.             1267
COPIED        10 MQ-Q-NAME-LENGTH              PIC S9(9) BINARY VALUE 48.             1268
COPIED        10 MQ-QSG-NAME-LENGTH            PIC S9(9) BINARY VALUE 4.              1269
COPIED        10 MQ-REMOTE-SYS-ID-LENGTH       PIC S9(9) BINARY VALUE 4.              1270
COPIED        10 MQ-SECURITY-ID-LENGTH         PIC S9(9) BINARY VALUE 40.             1271
COPIED        10 MQ-SERVICE-NAME-LENGTH        PIC S9(9) BINARY VALUE 32.             1272
```

Figure 22 MQGET Sample Compiled Listing (Page 16 of 35)

```
COPIED        10 MQ-SERVICE-STEP-LENGTH        PIC S9(9) BINARY VALUE 8.                          1273
COPIED        10 MQ-SHORT-CONN-NAME-LENGTH     PIC S9(9) BINARY VALUE 20.                         1274

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    27
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQ-SSL-CIPHER-SPEC-LENGTH     PIC S9(9) BINARY VALUE 32.                         1275
COPIED        10 MQ-SSL-CRYPTO-HARDWARE-LENGTH PIC S9(9) BINARY VALUE 256.                        1276
COPIED        10 MQ-SSL-HANDSHAKE-STAGE-LENGTH PIC S9(9) BINARY VALUE 32.                         1277
COPIED        10 MQ-SSL-KEY-REPOSITORY-LENGTH  PIC S9(9) BINARY VALUE 256.                        1278
COPIED        10 MQ-SSL-PEER-NAME-LENGTH       PIC S9(9) BINARY VALUE 1024.                       1279
COPIED        10 MQ-SSL-SHORT-PEER-NAME-LENGTH PIC S9(9) BINARY VALUE 256.                        1280
COPIED        10 MQ-START-CODE-LENGTH          PIC S9(9) BINARY VALUE 4.                          1281
COPIED        10 MQ-STORAGE-CLASS-DESC-LENGTH  PIC S9(9) BINARY VALUE 64.                         1282
COPIED        10 MQ-STORAGE-CLASS-LENGTH       PIC S9(9) BINARY VALUE 8.                          1283
COPIED        10 MQ-SUB-IDENTITY-LENGTH        PIC S9(9) BINARY VALUE 128.                        1284
COPIED        10 MQ-TIME-LENGTH                PIC S9(9) BINARY VALUE 8.                          1285
COPIED        10 MQ-TOTAL-EXIT-DATA-LENGTH     PIC S9(9) BINARY VALUE 999.                        1286
COPIED        10 MQ-TOTAL-EXIT-NAME-LENGTH     PIC S9(9) BINARY VALUE 999.                        1287
COPIED        10 MQ-TP-NAME-LENGTH             PIC S9(9) BINARY VALUE 64.                         1288
COPIED        10 MQ-TRAN-INSTANCE-ID-LENGTH    PIC S9(9) BINARY VALUE 16.                         1289
COPIED        10 MQ-TRANSACTION-ID-LENGTH      PIC S9(9) BINARY VALUE 4.                          1290
COPIED        10 MQ-TRIGGER-DATA-LENGTH        PIC S9(9) BINARY VALUE 64.                         1291
COPIED        10 MQ-USER-ID-LENGTH             PIC S9(9) BINARY VALUE 12.                         1292
COPIED        10 MQ-XCF-GROUP-NAME-LENGTH      PIC S9(9) BINARY VALUE 8.                          1293
COPIED        10 MQ-XCF-MEMBER-NAME-LENGTH     PIC S9(9) BINARY VALUE 16.                         1294
COPIED                                                                                            1295
COPIED      **   Completion Codes                                                                 1296
COPIED        10 MQCC-OK       PIC S9(9) BINARY VALUE 0.                                          1297
COPIED        10 MQCC-WARNING PIC S9(9) BINARY VALUE 1.                                           1298
COPIED        10 MQCC-FAILED  PIC S9(9) BINARY VALUE 2.                                           1299
COPIED        10 MQCC-UNKNOWN PIC S9(9) BINARY VALUE -1.                                          1300
COPIED                                                                                            1301
COPIED      **   Reason Codes                                                                     1302
COPIED        10 MQRC-NONE                     PIC S9(9) BINARY VALUE 0.                        1303
COPIED        10 MQRC-APPL-FIRST               PIC S9(9) BINARY VALUE 900.                      1304
COPIED        10 MQRC-APPL-LAST                PIC S9(9) BINARY VALUE 999.                      1305
COPIED        10 MQRC-ALIAS-BASE-Q-TYPE-ERROR  PIC S9(9) BINARY VALUE 2001.                     1306
COPIED        10 MQRC-ALREADY-CONNECTED        PIC S9(9) BINARY VALUE 2002.                     1307
COPIED        10 MQRC-BACKED-OUT               PIC S9(9) BINARY VALUE 2003.                     1308
COPIED        10 MQRC-BUFFER-ERROR             PIC S9(9) BINARY VALUE 2004.                     1309
COPIED        10 MQRC-BUFFER-LENGTH-ERROR      PIC S9(9) BINARY VALUE 2005.                     1310
COPIED        10 MQRC-CHAR-ATTR-LENGTH-ERROR   PIC S9(9) BINARY VALUE 2006.                     1311
COPIED        10 MQRC-CHAR-ATTRS-ERROR         PIC S9(9) BINARY VALUE 2007.                     1312
COPIED        10 MQRC-CHAR-ATTRS-TOO-SHORT     PIC S9(9) BINARY VALUE 2008.                     1313
COPIED        10 MQRC-CONNECTION-BROKEN        PIC S9(9) BINARY VALUE 2009.                     1314
COPIED        10 MQRC-DATA-LENGTH-ERROR        PIC S9(9) BINARY VALUE 2010.                     1315
COPIED        10 MQRC-DYNAMIC-Q-NAME-ERROR     PIC S9(9) BINARY VALUE 2011.                     1316
COPIED        10 MQRC-ENVIRONMENT-ERROR        PIC S9(9) BINARY VALUE 2012.                     1317
COPIED        10 MQRC-EXPIRY-ERROR             PIC S9(9) BINARY VALUE 2013.                     1318
COPIED        10 MQRC-FEEDBACK-ERROR           PIC S9(9) BINARY VALUE 2014.                     1319
COPIED        10 MQRC-GET-INHIBITED            PIC S9(9) BINARY VALUE 2016.                     1320
COPIED        10 MQRC-HANDLE-NOT-AVAILABLE     PIC S9(9) BINARY VALUE 2017.                     1321
COPIED        10 MQRC-HCONN-ERROR              PIC S9(9) BINARY VALUE 2018.                     1322
COPIED        10 MQRC-HOBJ-ERROR               PIC S9(9) BINARY VALUE 2019.                     1323

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    28
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQRC-INHIBIT-VALUE-ERROR      PIC S9(9) BINARY VALUE 2020.                     1324
COPIED        10 MQRC-INT-ATTR-COUNT-ERROR     PIC S9(9) BINARY VALUE 2021.                     1325
COPIED        10 MQRC-INT-ATTR-COUNT-TOO-SMALL PIC S9(9) BINARY VALUE 2022.                     1326
COPIED        10 MQRC-INT-ATTRS-ARRAY-ERROR    PIC S9(9) BINARY VALUE 2023.                     1327
COPIED        10 MQRC-SYNCPOINT-LIMIT-REACHED  PIC S9(9) BINARY VALUE 2024.                     1328
COPIED        10 MQRC-MAX-CONNS-LIMIT-REACHED  PIC S9(9) BINARY VALUE 2025.                     1329
COPIED        10 MQRC-MD-ERROR                 PIC S9(9) BINARY VALUE 2026.                     1330
COPIED        10 MQRC-MISSING-REPLY-TO-Q       PIC S9(9) BINARY VALUE 2027.                     1331
COPIED        10 MQRC-MSG-TYPE-ERROR           PIC S9(9) BINARY VALUE 2029.                     1332
COPIED        10 MQRC-MSG-TOO-BIG-FOR-Q        PIC S9(9) BINARY VALUE 2030.                     1333
COPIED        10 MQRC-MSG-TOO-BIG-FOR-Q-MGR    PIC S9(9) BINARY VALUE 2031.                     1334
COPIED        10 MQRC-NO-MSG-AVAILABLE         PIC S9(9) BINARY VALUE 2033.                     1335
COPIED        10 MQRC-NO-MSG-UNDER-CURSOR      PIC S9(9) BINARY VALUE 2034.                     1336
COPIED        10 MQRC-NOT-AUTHORIZED           PIC S9(9) BINARY VALUE 2035.                     1337
COPIED        10 MQRC-NOT-OPEN-FOR-BROWSE      PIC S9(9) BINARY VALUE 2036.                     1338
COPIED        10 MQRC-NOT-OPEN-FOR-INPUT       PIC S9(9) BINARY VALUE 2037.                     1339
COPIED        10 MQRC-NOT-OPEN-FOR-INQUIRE     PIC S9(9) BINARY VALUE 2038.                     1340
COPIED        10 MQRC-NOT-OPEN-FOR-OUTPUT      PIC S9(9) BINARY VALUE 2039.                     1341
COPIED        10 MQRC-NOT-OPEN-FOR-SET         PIC S9(9) BINARY VALUE 2040.                     1342
COPIED        10 MQRC-OBJECT-CHANGED           PIC S9(9) BINARY VALUE 2041.                     1343
COPIED        10 MQRC-OBJECT-IN-USE            PIC S9(9) BINARY VALUE 2042.                     1344
COPIED        10 MQRC-OBJECT-TYPE-ERROR        PIC S9(9) BINARY VALUE 2043.                     1345
COPIED        10 MQRC-OD-ERROR                 PIC S9(9) BINARY VALUE 2044.                     1346
COPIED        10 MQRC-OPTION-NOT-VALID-FOR-TYPE PIC S9(9) BINARY VALUE 2045.                    1347
COPIED        10 MQRC-OPTIONS-ERROR            PIC S9(9) BINARY VALUE 2046.                     1348
COPIED        10 MQRC-PERSISTENCE-ERROR        PIC S9(9) BINARY VALUE 2047.                     1349
COPIED        10 MQRC-PERSISTENT-NOT-ALLOWED   PIC S9(9) BINARY VALUE 2048.                     1350
COPIED        10 MQRC-PRIORITY-EXCEEDS-MAXIMUM PIC S9(9) BINARY VALUE 2049.                     1351
```

Figure 22 MQGET Sample Compiled Listing (Page 17 of 35)

```
COPIED          10 MQRC-PRIORITY-ERROR           PIC S9(9) BINARY VALUE 2050.                    1352
COPIED          10 MQRC-PUT-INHIBITED            PIC S9(9) BINARY VALUE 2051.                    1353
COPIED          10 MQRC-Q-DELETED                PIC S9(9) BINARY VALUE 2052.                    1354
COPIED          10 MQRC-Q-FULL                   PIC S9(9) BINARY VALUE 2053.                    1355
COPIED          10 MQRC-Q-NOT-EMPTY              PIC S9(9) BINARY VALUE 2055.                    1356
COPIED          10 MQRC-Q-SPACE-NOT-AVAILABLE    PIC S9(9) BINARY VALUE 2056.                    1357
COPIED          10 MQRC-Q-TYPE-ERROR             PIC S9(9) BINARY VALUE 2057.                    1358
COPIED          10 MQRC-Q-MGR-NAME-ERROR         PIC S9(9) BINARY VALUE 2058.                    1359
COPIED          10 MQRC-Q-MGR-NOT-AVAILABLE      PIC S9(9) BINARY VALUE 2059.                    1360
COPIED          10 MQRC-REPORT-OPTIONS-ERROR     PIC S9(9) BINARY VALUE 2061.                    1361
COPIED          10 MQRC-SECOND-MARK-NOT-ALLOWED  PIC S9(9) BINARY VALUE 2062.                    1362
COPIED          10 MQRC-SECURITY-ERROR           PIC S9(9) BINARY VALUE 2063.                    1363
COPIED          10 MQRC-SELECTOR-COUNT-ERROR     PIC S9(9) BINARY VALUE 2065.                    1364
COPIED          10 MQRC-SELECTOR-LIMIT-EXCEEDED  PIC S9(9) BINARY VALUE 2066.                    1365
COPIED          10 MQRC-SELECTOR-ERROR           PIC S9(9) BINARY VALUE 2067.                    1366
COPIED          10 MQRC-SELECTOR-NOT-FOR-TYPE    PIC S9(9) BINARY VALUE 2068.                    1367
COPIED          10 MQRC-SIGNAL-OUTSTANDING       PIC S9(9) BINARY VALUE 2069.                    1368
COPIED          10 MQRC-SIGNAL-REQUEST-ACCEPTED  PIC S9(9) BINARY VALUE 2070.                    1369
COPIED          10 MQRC-STORAGE-NOT-AVAILABLE    PIC S9(9) BINARY VALUE 2071.                    1370
COPIED          10 MQRC-SYNCPOINT-NOT-AVAILABLE  PIC S9(9) BINARY VALUE 2072.                    1371
COPIED          10 MQRC-TRIGGER-CONTROL-ERROR    PIC S9(9) BINARY VALUE 2075.                    1372

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    29
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED          10 MQRC-TRIGGER-DEPTH-ERROR      PIC S9(9) BINARY VALUE 2076.                    1373
COPIED          10 MQRC-TRIGGER-MSG-PRIORITY-ERR PIC S9(9) BINARY VALUE 2077.                    1374
COPIED          10 MQRC-TRIGGER-TYPE-ERROR       PIC S9(9) BINARY VALUE 2078.                    1375
COPIED          10 MQRC-TRUNCATED-MSG-ACCEPTED   PIC S9(9) BINARY VALUE 2079.                    1376
COPIED          10 MQRC-TRUNCATED-MSG-FAILED     PIC S9(9) BINARY VALUE 2080.                    1377
COPIED          10 MQRC-UNKNOWN-ALIAS-BASE-Q     PIC S9(9) BINARY VALUE 2082.                    1378
COPIED          10 MQRC-UNKNOWN-OBJECT-NAME      PIC S9(9) BINARY VALUE 2085.                    1379
COPIED          10 MQRC-UNKNOWN-OBJECT-Q-MGR     PIC S9(9) BINARY VALUE 2086.                    1380
COPIED          10 MQRC-UNKNOWN-REMOTE-Q-MGR     PIC S9(9) BINARY VALUE 2087.                    1381
COPIED          10 MQRC-WAIT-INTERVAL-ERROR      PIC S9(9) BINARY VALUE 2090.                    1382
COPIED          10 MQRC-XMIT-Q-TYPE-ERROR        PIC S9(9) BINARY VALUE 2091.                    1383
COPIED          10 MQRC-XMIT-Q-USAGE-ERROR       PIC S9(9) BINARY VALUE 2092.                    1384
COPIED          10 MQRC-NOT-OPEN-FOR-PASS-ALL    PIC S9(9) BINARY VALUE 2093.                    1385
COPIED          10 MQRC-NOT-OPEN-FOR-PASS-IDENT  PIC S9(9) BINARY VALUE 2094.                    1386
COPIED          10 MQRC-NOT-OPEN-FOR-SET-ALL     PIC S9(9) BINARY VALUE 2095.                    1387
COPIED          10 MQRC-NOT-OPEN-FOR-SET-IDENT   PIC S9(9) BINARY VALUE 2096.                    1388
COPIED          10 MQRC-CONTEXT-HANDLE-ERROR     PIC S9(9) BINARY VALUE 2097.                    1389
COPIED          10 MQRC-CONTEXT-NOT-AVAILABLE    PIC S9(9) BINARY VALUE 2098.                    1390
COPIED          10 MQRC-SIGNAL1-ERROR            PIC S9(9) BINARY VALUE 2099.                    1391
COPIED          10 MQRC-OBJECT-ALREADY-EXISTS    PIC S9(9) BINARY VALUE 2100.                    1392
COPIED          10 MQRC-OBJECT-DAMAGED           PIC S9(9) BINARY VALUE 2101.                    1393
COPIED          10 MQRC-RESOURCE-PROBLEM         PIC S9(9) BINARY VALUE 2102.                    1394
COPIED          10 MQRC-ANOTHER-Q-MGR-CONNECTED  PIC S9(9) BINARY VALUE 2103.                    1395
COPIED          10 MQRC-UNKNOWN-REPORT-OPTION    PIC S9(9) BINARY VALUE 2104.                    1396
COPIED          10 MQRC-STORAGE-CLASS-ERROR      PIC S9(9) BINARY VALUE 2105.                    1397
COPIED          10 MQRC-COD-NOT-VALID-FOR-XCF-Q  PIC S9(9) BINARY VALUE 2106.                    1398
COPIED          10 MQRC-SUPPRESSED-BY-EXIT       PIC S9(9) BINARY VALUE 2109.                    1399
COPIED          10 MQRC-FORMAT-ERROR             PIC S9(9) BINARY VALUE 2110.                    1400
COPIED          10 MQRC-SOURCE-CCSID-ERROR       PIC S9(9) BINARY VALUE 2111.                    1401
COPIED          10 MQRC-SOURCE-INTEGER-ENC-ERROR PIC S9(9) BINARY VALUE 2112.                    1402
COPIED          10 MQRC-SOURCE-DECIMAL-ENC-ERROR PIC S9(9) BINARY VALUE 2113.                    1403
COPIED          10 MQRC-SOURCE-FLOAT-ENC-ERROR   PIC S9(9) BINARY VALUE 2114.                    1404
COPIED          10 MQRC-TARGET-CCSID-ERROR       PIC S9(9) BINARY VALUE 2115.                    1405
COPIED          10 MQRC-TARGET-INTEGER-ENC-ERROR PIC S9(9) BINARY VALUE 2116.                    1406
COPIED          10 MQRC-TARGET-DECIMAL-ENC-ERROR PIC S9(9) BINARY VALUE 2117.                    1407
COPIED          10 MQRC-TARGET-FLOAT-ENC-ERROR   PIC S9(9) BINARY VALUE 2118.                    1408
COPIED          10 MQRC-NOT-CONVERTED            PIC S9(9) BINARY VALUE 2119.                    1409
COPIED          10 MQRC-CONVERTED-MSG-TOO-BIG    PIC S9(9) BINARY VALUE 2120.                    1410
COPIED          10 MQRC-TRUNCATED                PIC S9(9) BINARY VALUE 2120.                    1411
COPIED          10 MQRC-NO-EXTERNAL-PARTICIPANTS PIC S9(9) BINARY VALUE 2121.                    1412
COPIED          10 MQRC-PARTICIPANT-NOT-AVAILABLE PIC S9(9) BINARY VALUE 2122.                   1413
COPIED          10 MQRC-OUTCOME-MIXED            PIC S9(9) BINARY VALUE 2123.                    1414
COPIED          10 MQRC-OUTCOME-PENDING          PIC S9(9) BINARY VALUE 2124.                    1415
COPIED          10 MQRC-BRIDGE-STARTED           PIC S9(9) BINARY VALUE 2125.                    1416
COPIED          10 MQRC-BRIDGE-STOPPED           PIC S9(9) BINARY VALUE 2126.                    1417
COPIED          10 MQRC-ADAPTER-STORAGE-SHORTAGE PIC S9(9) BINARY VALUE 2127.                    1418
COPIED          10 MQRC-UOW-IN-PROGRESS          PIC S9(9) BINARY VALUE 2128.                    1419
COPIED          10 MQRC-ADAPTER-CONN-LOAD-ERROR  PIC S9(9) BINARY VALUE 2129.                    1420
COPIED          10 MQRC-ADAPTER-SERV-LOAD-ERROR  PIC S9(9) BINARY VALUE 2130.                    1421

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    30
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED          10 MQRC-ADAPTER-DEFS-ERROR       PIC S9(9) BINARY VALUE 2131.                    1422
COPIED          10 MQRC-ADAPTER-DEFS-LOAD-ERROR  PIC S9(9) BINARY VALUE 2132.                    1423
COPIED          10 MQRC-ADAPTER-CONV-LOAD-ERROR  PIC S9(9) BINARY VALUE 2133.                    1424
COPIED          10 MQRC-BO-ERROR                 PIC S9(9) BINARY VALUE 2134.                    1425
COPIED          10 MQRC-DH-ERROR                 PIC S9(9) BINARY VALUE 2135.                    1426
COPIED          10 MQRC-MULTIPLE-REASONS         PIC S9(9) BINARY VALUE 2136.                    1427
COPIED          10 MQRC-OPEN-FAILED              PIC S9(9) BINARY VALUE 2137.                    1428
COPIED          10 MQRC-ADAPTER-DISC-LOAD-ERROR  PIC S9(9) BINARY VALUE 2138.                    1429
COPIED          10 MQRC-CNO-ERROR                PIC S9(9) BINARY VALUE 2139.                    1430
```

Figure 22 MQGET Sample Compiled Listing (Page 18 of 35)

```
COPIED          10 MQRC-CICS-WAIT-FAILED          PIC S9(9) BINARY VALUE 2140.                          1431
COPIED          10 MQRC-DLH-ERROR                 PIC S9(9) BINARY VALUE 2141.                          1432
COPIED          10 MQRC-HEADER-ERROR              PIC S9(9) BINARY VALUE 2142.                          1433
COPIED          10 MQRC-SOURCE-LENGTH-ERROR       PIC S9(9) BINARY VALUE 2143.                          1434
COPIED          10 MQRC-TARGET-LENGTH-ERROR       PIC S9(9) BINARY VALUE 2144.                          1435
COPIED          10 MQRC-SOURCE-BUFFER-ERROR       PIC S9(9) BINARY VALUE 2145.                          1436
COPIED          10 MQRC-TARGET-BUFFER-ERROR       PIC S9(9) BINARY VALUE 2146.                          1437
COPIED          10 MQRC-IIH-ERROR                 PIC S9(9) BINARY VALUE 2148.                          1438
COPIED          10 MQRC-PCF-ERROR                 PIC S9(9) BINARY VALUE 2149.                          1439
COPIED          10 MQRC-DBCS-ERROR                PIC S9(9) BINARY VALUE 2150.                          1440
COPIED          10 MQRC-OBJECT-NAME-ERROR         PIC S9(9) BINARY VALUE 2152.                          1441
COPIED          10 MQRC-OBJECT-Q-MGR-NAME-ERROR   PIC S9(9) BINARY VALUE 2153.                          1442
COPIED          10 MQRC-RECS-PRESENT-ERROR        PIC S9(9) BINARY VALUE 2154.                          1443
COPIED          10 MQRC-OBJECT-RECORDS-ERROR      PIC S9(9) BINARY VALUE 2155.                          1444
COPIED          10 MQRC-RESPONSE-RECORDS-ERROR    PIC S9(9) BINARY VALUE 2156.                          1445
COPIED          10 MQRC-ASID-MISMATCH             PIC S9(9) BINARY VALUE 2157.                          1446
COPIED          10 MQRC-PMO-RECORD-FLAGS-ERROR    PIC S9(9) BINARY VALUE 2158.                          1447
COPIED          10 MQRC-PUT-MSG-RECORDS-ERROR     PIC S9(9) BINARY VALUE 2159.                          1448
COPIED          10 MQRC-CONN-ID-IN-USE            PIC S9(9) BINARY VALUE 2160.                          1449
COPIED          10 MQRC-Q-MGR-QUIESCING           PIC S9(9) BINARY VALUE 2161.                          1450
COPIED          10 MQRC-Q-MGR-STOPPING            PIC S9(9) BINARY VALUE 2162.                          1451
COPIED          10 MQRC-DUPLICATE-RECOV-COORD     PIC S9(9) BINARY VALUE 2163.                          1452
COPIED          10 MQRC-PMO-ERROR                 PIC S9(9) BINARY VALUE 2173.                          1453
COPIED          10 MQRC-API-EXIT-NOT-FOUND        PIC S9(9) BINARY VALUE 2182.                          1454
COPIED          10 MQRC-API-EXIT-LOAD-ERROR       PIC S9(9) BINARY VALUE 2183.                          1455
COPIED          10 MQRC-REMOTE-Q-NAME-ERROR       PIC S9(9) BINARY VALUE 2184.                          1456
COPIED          10 MQRC-INCONSISTENT-PERSISTENCE  PIC S9(9) BINARY VALUE 2185.                          1457
COPIED          10 MQRC-GMO-ERROR                 PIC S9(9) BINARY VALUE 2186.                          1458
COPIED          10 MQRC-CICS-BRIDGE-RESTRICTION   PIC S9(9) BINARY VALUE 2187.                          1459
COPIED          10 MQRC-STOPPED-BY-CLUSTER-EXIT   PIC S9(9) BINARY VALUE 2188.                          1460
COPIED          10 MQRC-CLUSTER-RESOLUTION-ERROR  PIC S9(9) BINARY VALUE 2189.                          1461
COPIED          10 MQRC-CONVERTED-STRING-TOO-BIG  PIC S9(9) BINARY VALUE 2190.                          1462
COPIED          10 MQRC-TMC-ERROR                 PIC S9(9) BINARY VALUE 2191.                          1463
COPIED          10 MQRC-PAGESET-FULL              PIC S9(9) BINARY VALUE 2192.                          1464
COPIED          10 MQRC-STORAGE-MEDIUM-FULL       PIC S9(9) BINARY VALUE 2192.                          1465
COPIED          10 MQRC-PAGESET-ERROR             PIC S9(9) BINARY VALUE 2193.                          1466
COPIED          10 MQRC-NAME-NOT-VALID-FOR-TYPE   PIC S9(9) BINARY VALUE 2194.                          1467
COPIED          10 MQRC-UNEXPECTED-ERROR          PIC S9(9) BINARY VALUE 2195.                          1468
COPIED          10 MQRC-UNKNOWN-XMIT-Q            PIC S9(9) BINARY VALUE 2196.                          1469
COPIED          10 MQRC-UNKNOWN-DEF-XMIT-Q        PIC S9(9) BINARY VALUE 2197.                          1470

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    31
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED          10 MQRC-DEF-XMIT-Q-TYPE-ERROR     PIC S9(9) BINARY VALUE 2198.                          1471
COPIED          10 MQRC-DEF-XMIT-Q-USAGE-ERROR    PIC S9(9) BINARY VALUE 2199.                          1472
COPIED          10 MQRC-NAME-IN-USE               PIC S9(9) BINARY VALUE 2201.                          1473
COPIED          10 MQRC-CONNECTION-QUIESCING      PIC S9(9) BINARY VALUE 2202.                          1474
COPIED          10 MQRC-CONNECTION-STOPPING       PIC S9(9) BINARY VALUE 2203.                          1475
COPIED          10 MQRC-ADAPTER-NOT-AVAILABLE     PIC S9(9) BINARY VALUE 2204.                          1476
COPIED          10 MQRC-MSG-ID-ERROR              PIC S9(9) BINARY VALUE 2206.                          1477
COPIED          10 MQRC-CORREL-ID-ERROR           PIC S9(9) BINARY VALUE 2207.                          1478
COPIED          10 MQRC-FILE-SYSTEM-ERROR         PIC S9(9) BINARY VALUE 2208.                          1479
COPIED          10 MQRC-NO-MSG-LOCKED             PIC S9(9) BINARY VALUE 2209.                          1480
COPIED          10 MQRC-FILE-NOT-AUDITED          PIC S9(9) BINARY VALUE 2216.                          1481
COPIED          10 MQRC-CONNECTION-NOT-AUTHORIZED PIC S9(9) BINARY VALUE 2217.                          1482
COPIED          10 MQRC-MSG-TOO-BIG-FOR-CHANNEL   PIC S9(9) BINARY VALUE 2218.                          1483
COPIED          10 MQRC-CALL-IN-PROGRESS          PIC S9(9) BINARY VALUE 2219.                          1484
COPIED          10 MQRC-RMH-ERROR                 PIC S9(9) BINARY VALUE 2220.                          1485
COPIED          10 MQRC-Q-MGR-ACTIVE              PIC S9(9) BINARY VALUE 2222.                          1486
COPIED          10 MQRC-Q-MGR-NOT-ACTIVE          PIC S9(9) BINARY VALUE 2223.                          1487
COPIED          10 MQRC-Q-DEPTH-HIGH              PIC S9(9) BINARY VALUE 2224.                          1488
COPIED          10 MQRC-Q-DEPTH-LOW               PIC S9(9) BINARY VALUE 2225.                          1489
COPIED          10 MQRC-Q-SERVICE-INTERVAL-HIGH   PIC S9(9) BINARY VALUE 2226.                          1490
COPIED          10 MQRC-Q-SERVICE-INTERVAL-OK     PIC S9(9) BINARY VALUE 2227.                          1491
COPIED          10 MQRC-UNIT-OF-WORK-NOT-STARTED  PIC S9(9) BINARY VALUE 2232.                          1492
COPIED          10 MQRC-CHANNEL-AUTO-DEF-OK       PIC S9(9) BINARY VALUE 2233.                          1493
COPIED          10 MQRC-CHANNEL-AUTO-DEF-ERROR    PIC S9(9) BINARY VALUE 2234.                          1494
COPIED          10 MQRC-CFH-ERROR                 PIC S9(9) BINARY VALUE 2235.                          1495
COPIED          10 MQRC-CFIL-ERROR                PIC S9(9) BINARY VALUE 2236.                          1496
COPIED          10 MQRC-CFIN-ERROR                PIC S9(9) BINARY VALUE 2237.                          1497
COPIED          10 MQRC-CFSL-ERROR                PIC S9(9) BINARY VALUE 2238.                          1498
COPIED          10 MQRC-CFST-ERROR                PIC S9(9) BINARY VALUE 2239.                          1499
COPIED          10 MQRC-INCOMPLETE-GROUP          PIC S9(9) BINARY VALUE 2241.                          1500
COPIED          10 MQRC-INCOMPLETE-MSG            PIC S9(9) BINARY VALUE 2242.                          1501
COPIED          10 MQRC-INCONSISTENT-CCSIDS       PIC S9(9) BINARY VALUE 2243.                          1502
COPIED          10 MQRC-INCONSISTENT-ENCODINGS    PIC S9(9) BINARY VALUE 2244.                          1503
COPIED          10 MQRC-INCONSISTENT-UOW          PIC S9(9) BINARY VALUE 2245.                          1504
COPIED          10 MQRC-INVALID-MSG-UNDER-CURSOR  PIC S9(9) BINARY VALUE 2246.                          1505
COPIED          10 MQRC-MATCH-OPTIONS-ERROR       PIC S9(9) BINARY VALUE 2247.                          1506
COPIED          10 MQRC-MDE-ERROR                 PIC S9(9) BINARY VALUE 2248.                          1507
COPIED          10 MQRC-MSG-FLAGS-ERROR           PIC S9(9) BINARY VALUE 2249.                          1508
COPIED          10 MQRC-MSG-SEQ-NUMBER-ERROR      PIC S9(9) BINARY VALUE 2250.                          1509
COPIED          10 MQRC-OFFSET-ERROR              PIC S9(9) BINARY VALUE 2251.                          1510
COPIED          10 MQRC-ORIGINAL-LENGTH-ERROR     PIC S9(9) BINARY VALUE 2252.                          1511
COPIED          10 MQRC-SEGMENT-LENGTH-ZERO       PIC S9(9) BINARY VALUE 2253.                          1512
```

Figure 22 MQGET Sample Compiled Listing (Page 19 of 35)

```
COPIED         10 MQRC-UOW-NOT-AVAILABLE          PIC S9(9) BINARY VALUE 2255.                              1513
COPIED         10 MQRC-WRONG-GMO-VERSION          PIC S9(9) BINARY VALUE 2256.                              1514
COPIED         10 MQRC-WRONG-MD-VERSION           PIC S9(9) BINARY VALUE 2257.                              1515
COPIED         10 MQRC-GROUP-ID-ERROR             PIC S9(9) BINARY VALUE 2258.                              1516
COPIED         10 MQRC-INCONSISTENT-BROWSE        PIC S9(9) BINARY VALUE 2259.                              1517
COPIED         10 MQRC-XQH-ERROR                  PIC S9(9) BINARY VALUE 2260.                              1518
COPIED         10 MQRC-SRC-ENV-ERROR              PIC S9(9) BINARY VALUE 2261.                              1519

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    32
       1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED         10 MQRC-SRC-NAME-ERROR             PIC S9(9) BINARY VALUE 2262.                              1520
COPIED         10 MQRC-DEST-ENV-ERROR             PIC S9(9) BINARY VALUE 2263.                              1521
COPIED         10 MQRC-DEST-NAME-ERROR            PIC S9(9) BINARY VALUE 2264.                              1522
COPIED         10 MQRC-TM-ERROR                   PIC S9(9) BINARY VALUE 2265.                              1523
COPIED         10 MQRC-CLUSTER-EXIT-ERROR         PIC S9(9) BINARY VALUE 2266.                              1524
COPIED         10 MQRC-CLUSTER-EXIT-LOAD-ERROR    PIC S9(9) BINARY VALUE 2267.                              1525
COPIED         10 MQRC-CLUSTER-PUT-INHIBITED      PIC S9(9) BINARY VALUE 2268.                              1526
COPIED         10 MQRC-CLUSTER-RESOURCE-ERROR     PIC S9(9) BINARY VALUE 2269.                              1527
COPIED         10 MQRC-NO-DESTINATIONS-AVAILABLE  PIC S9(9) BINARY VALUE 2270.                              1528
COPIED         10 MQRC-CONN-TAG-IN-USE            PIC S9(9) BINARY VALUE 2271.                              1529
COPIED         10 MQRC-PARTIALLY-CONVERTED        PIC S9(9) BINARY VALUE 2272.                              1530
COPIED         10 MQRC-CONNECTION-ERROR           PIC S9(9) BINARY VALUE 2273.                              1531
COPIED         10 MQRC-OPTION-ENVIRONMENT-ERROR   PIC S9(9) BINARY VALUE 2274.                              1532
COPIED         10 MQRC-CD-ERROR                   PIC S9(9) BINARY VALUE 2277.                              1533
COPIED         10 MQRC-CLIENT-CONN-ERROR          PIC S9(9) BINARY VALUE 2278.                              1534
COPIED         10 MQRC-CHANNEL-STOPPED-BY-USER    PIC S9(9) BINARY VALUE 2279.                              1535
COPIED         10 MQRC-HCONFIG-ERROR              PIC S9(9) BINARY VALUE 2280.                              1536
COPIED         10 MQRC-FUNCTION-ERROR             PIC S9(9) BINARY VALUE 2281.                              1537
COPIED         10 MQRC-CHANNEL-STARTED            PIC S9(9) BINARY VALUE 2282.                              1538
COPIED         10 MQRC-CHANNEL-STOPPED            PIC S9(9) BINARY VALUE 2283.                              1539
COPIED         10 MQRC-CHANNEL-CONV-ERROR         PIC S9(9) BINARY VALUE 2284.                              1540
COPIED         10 MQRC-SERVICE-NOT-AVAILABLE      PIC S9(9) BINARY VALUE 2285.                              1541
COPIED         10 MQRC-INITIALIZATION-FAILED      PIC S9(9) BINARY VALUE 2286.                              1542
COPIED         10 MQRC-TERMINATION-FAILED         PIC S9(9) BINARY VALUE 2287.                              1543
COPIED         10 MQRC-UNKNOWN-Q-NAME             PIC S9(9) BINARY VALUE 2288.                              1544
COPIED         10 MQRC-SERVICE-ERROR              PIC S9(9) BINARY VALUE 2289.                              1545
COPIED         10 MQRC-Q-ALREADY-EXISTS           PIC S9(9) BINARY VALUE 2290.                              1546
COPIED         10 MQRC-USER-ID-NOT-AVAILABLE      PIC S9(9) BINARY VALUE 2291.                              1547
COPIED         10 MQRC-UNKNOWN-ENTITY             PIC S9(9) BINARY VALUE 2292.                              1548
COPIED         10 MQRC-UNKNOWN-AUTH-ENTITY        PIC S9(9) BINARY VALUE 2293.                              1549
COPIED         10 MQRC-UNKNOWN-REF-OBJECT         PIC S9(9) BINARY VALUE 2294.                              1550
COPIED         10 MQRC-CHANNEL-ACTIVATED          PIC S9(9) BINARY VALUE 2295.                              1551
COPIED         10 MQRC-CHANNEL-NOT-ACTIVATED      PIC S9(9) BINARY VALUE 2296.                              1552
COPIED         10 MQRC-UOW-CANCELED               PIC S9(9) BINARY VALUE 2297.                              1553
COPIED         10 MQRC-FUNCTION-NOT-SUPPORTED     PIC S9(9) BINARY VALUE 2298.                              1554
COPIED         10 MQRC-SELECTOR-TYPE-ERROR        PIC S9(9) BINARY VALUE 2299.                              1555
COPIED         10 MQRC-COMMAND-TYPE-ERROR         PIC S9(9) BINARY VALUE 2300.                              1556
COPIED         10 MQRC-MULTIPLE-INSTANCE-ERROR    PIC S9(9) BINARY VALUE 2301.                              1557
COPIED         10 MQRC-SYSTEM-ITEM-NOT-ALTERABLE  PIC S9(9) BINARY VALUE 2302.                              1558
COPIED         10 MQRC-BAG-CONVERSION-ERROR       PIC S9(9) BINARY VALUE 2303.                              1559
COPIED         10 MQRC-SELECTOR-OUT-OF-RANGE      PIC S9(9) BINARY VALUE 2304.                              1560
COPIED         10 MQRC-SELECTOR-NOT-UNIQUE        PIC S9(9) BINARY VALUE 2305.                              1561
COPIED         10 MQRC-INDEX-NOT-PRESENT          PIC S9(9) BINARY VALUE 2306.                              1562
COPIED         10 MQRC-STRING-ERROR               PIC S9(9) BINARY VALUE 2307.                              1563
COPIED         10 MQRC-ENCODING-NOT-SUPPORTED     PIC S9(9) BINARY VALUE 2308.                              1564
COPIED         10 MQRC-SELECTOR-NOT-PRESENT       PIC S9(9) BINARY VALUE 2309.                              1565
COPIED         10 MQRC-OUT-SELECTOR-ERROR         PIC S9(9) BINARY VALUE 2310.                              1566
COPIED         10 MQRC-STRING-TRUNCATED           PIC S9(9) BINARY VALUE 2311.                              1567
COPIED         10 MQRC-SELECTOR-WRONG-TYPE        PIC S9(9) BINARY VALUE 2312.                              1568

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    33
       1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED         10 MQRC-INCONSISTENT-ITEM-TYPE     PIC S9(9) BINARY VALUE 2313.                              1569
COPIED         10 MQRC-INDEX-ERROR                PIC S9(9) BINARY VALUE 2314.                              1570
COPIED         10 MQRC-SYSTEM-BAG-NOT-ALTERABLE   PIC S9(9) BINARY VALUE 2315.                              1571
COPIED         10 MQRC-ITEM-COUNT-ERROR           PIC S9(9) BINARY VALUE 2316.                              1572
COPIED         10 MQRC-FORMAT-NOT-SUPPORTED       PIC S9(9) BINARY VALUE 2317.                              1573
COPIED         10 MQRC-SELECTOR-NOT-SUPPORTED     PIC S9(9) BINARY VALUE 2318.                              1574
COPIED         10 MQRC-ITEM-VALUE-ERROR           PIC S9(9) BINARY VALUE 2319.                              1575
COPIED         10 MQRC-HBAG-ERROR                 PIC S9(9) BINARY VALUE 2320.                              1576
COPIED         10 MQRC-PARAMETER-MISSING          PIC S9(9) BINARY VALUE 2321.                              1577
COPIED         10 MQRC-CMD-SERVER-NOT-AVAILABLE   PIC S9(9) BINARY VALUE 2322.                              1578
COPIED         10 MQRC-STRING-LENGTH-ERROR        PIC S9(9) BINARY VALUE 2323.                              1579
COPIED         10 MQRC-INQUIRY-COMMAND-ERROR      PIC S9(9) BINARY VALUE 2324.                              1580
COPIED         10 MQRC-NESTED-BAG-NOT-SUPPORTED   PIC S9(9) BINARY VALUE 2325.                              1581
COPIED         10 MQRC-BAG-WRONG-TYPE             PIC S9(9) BINARY VALUE 2326.                              1582
COPIED         10 MQRC-ITEM-TYPE-ERROR            PIC S9(9) BINARY VALUE 2327.                              1583
COPIED         10 MQRC-SYSTEM-BAG-NOT-DELETABLE   PIC S9(9) BINARY VALUE 2328.                              1584
COPIED         10 MQRC-SYSTEM-ITEM-NOT-DELETABLE  PIC S9(9) BINARY VALUE 2329.                              1585
COPIED         10 MQRC-CODED-CHAR-SET-ID-ERROR    PIC S9(9) BINARY VALUE 2330.                              1586
COPIED         10 MQRC-MSG-TOKEN-ERROR            PIC S9(9) BINARY VALUE 2331.                              1587
COPIED         10 MQRC-MISSING-WIH                PIC S9(9) BINARY VALUE 2332.                              1588
COPIED         10 MQRC-WIH-ERROR                  PIC S9(9) BINARY VALUE 2333.                              1589
COPIED         10 MQRC-RFH-ERROR                  PIC S9(9) BINARY VALUE 2334.                              1590
COPIED         10 MQRC-RFH-STRING-ERROR           PIC S9(9) BINARY VALUE 2335.                              1591
```

Figure 22 MQGET Sample Compiled Listing (Page 20 of 35)

```
COPIED          10 MQRC-RFH-COMMAND-ERROR          PIC S9(9) BINARY VALUE 2336.                          1592
COPIED          10 MQRC-RFH-PARM-ERROR             PIC S9(9) BINARY VALUE 2337.                          1593
COPIED          10 MQRC-RFH-DUPLICATE-PARM         PIC S9(9) BINARY VALUE 2338.                          1594
COPIED          10 MQRC-RFH-PARM-MISSING           PIC S9(9) BINARY VALUE 2339.                          1595
COPIED          10 MQRC-CHAR-CONVERSION-ERROR      PIC S9(9) BINARY VALUE 2340.                          1596
COPIED          10 MQRC-UCS2-CONVERSION-ERROR      PIC S9(9) BINARY VALUE 2341.                          1597
COPIED          10 MQRC-DB2-NOT-AVAILABLE          PIC S9(9) BINARY VALUE 2342.                          1598
COPIED          10 MQRC-OBJECT-NOT-UNIQUE          PIC S9(9) BINARY VALUE 2343.                          1599
COPIED          10 MQRC-CONN-TAG-NOT-RELEASED      PIC S9(9) BINARY VALUE 2344.                          1600
COPIED          10 MQRC-CF-NOT-AVAILABLE           PIC S9(9) BINARY VALUE 2345.                          1601
COPIED          10 MQRC-CF-STRUC-IN-USE            PIC S9(9) BINARY VALUE 2346.                          1602
COPIED          10 MQRC-CF-STRUC-LIST-HDR-IN-USE   PIC S9(9) BINARY VALUE 2347.                          1603
COPIED          10 MQRC-CF-STRUC-AUTH-FAILED       PIC S9(9) BINARY VALUE 2348.                          1604
COPIED          10 MQRC-CF-STRUC-ERROR             PIC S9(9) BINARY VALUE 2349.                          1605
COPIED          10 MQRC-CONN-TAG-NOT-USABLE        PIC S9(9) BINARY VALUE 2350.                          1606
COPIED          10 MQRC-GLOBAL-UOW-CONFLICT        PIC S9(9) BINARY VALUE 2351.                          1607
COPIED          10 MQRC-LOCAL-UOW-CONFLICT         PIC S9(9) BINARY VALUE 2352.                          1608
COPIED          10 MQRC-HANDLE-IN-USE-FOR-UOW      PIC S9(9) BINARY VALUE 2353.                          1609
COPIED          10 MQRC-UOW-ENLISTMENT-ERROR       PIC S9(9) BINARY VALUE 2354.                          1610
COPIED          10 MQRC-UOW-MIX-NOT-SUPPORTED      PIC S9(9) BINARY VALUE 2355.                          1611
COPIED          10 MQRC-WXP-ERROR                  PIC S9(9) BINARY VALUE 2356.                          1612
COPIED          10 MQRC-CURRENT-RECORD-ERROR       PIC S9(9) BINARY VALUE 2357.                          1613
COPIED          10 MQRC-NEXT-OFFSET-ERROR          PIC S9(9) BINARY VALUE 2358.                          1614
COPIED          10 MQRC-NO-RECORD-AVAILABLE        PIC S9(9) BINARY VALUE 2359.                          1615
COPIED          10 MQRC-OBJECT-LEVEL-INCOMPATIBLE  PIC S9(9) BINARY VALUE 2360.                          1616
COPIED          10 MQRC-NEXT-RECORD-ERROR          PIC S9(9) BINARY VALUE 2361.                          1617

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    34
       1--------------------VISION:RESULTS FREE FORM TEXT----------------------72         --------
COPIED          10 MQRC-BACKOUT-THRESHOLD-REACHED  PIC S9(9) BINARY VALUE 2362.                          1618
COPIED          10 MQRC-MSG-NOT-MATCHED            PIC S9(9) BINARY VALUE 2363.                          1619
COPIED          10 MQRC-JMS-FORMAT-ERROR           PIC S9(9) BINARY VALUE 2364.                          1620
COPIED          10 MQRC-SEGMENTS-NOT-SUPPORTED     PIC S9(9) BINARY VALUE 2365.                          1621
COPIED          10 MQRC-WRONG-CF-LEVEL             PIC S9(9) BINARY VALUE 2366.                          1622
COPIED          10 MQRC-CONFIG-CREATE-OBJECT       PIC S9(9) BINARY VALUE 2367.                          1623
COPIED          10 MQRC-CONFIG-CHANGE-OBJECT       PIC S9(9) BINARY VALUE 2368.                          1624
COPIED          10 MQRC-CONFIG-DELETE-OBJECT       PIC S9(9) BINARY VALUE 2369.                          1625
COPIED          10 MQRC-CONFIG-REFRESH-OBJECT      PIC S9(9) BINARY VALUE 2370.                          1626
COPIED          10 MQRC-CHANNEL-SSL-ERROR          PIC S9(9) BINARY VALUE 2371.                          1627
COPIED          10 MQRC-API-EXIT-ERROR             PIC S9(9) BINARY VALUE 2374.                          1628
COPIED          10 MQRC-API-EXIT-INIT-ERROR        PIC S9(9) BINARY VALUE 2375.                          1629
COPIED          10 MQRC-API-EXIT-TERM-ERROR        PIC S9(9) BINARY VALUE 2376.                          1630
COPIED          10 MQRC-EXIT-REASON-ERROR          PIC S9(9) BINARY VALUE 2377.                          1631
COPIED          10 MQRC-RESERVED-VALUE-ERROR       PIC S9(9) BINARY VALUE 2378.                          1632
COPIED          10 MQRC-NO-DATA-AVAILABLE          PIC S9(9) BINARY VALUE 2379.                          1633
COPIED          10 MQRC-SCO-ERROR                  PIC S9(9) BINARY VALUE 2380.                          1634
COPIED          10 MQRC-KEY-REPOSITORY-ERROR       PIC S9(9) BINARY VALUE 2381.                          1635
COPIED          10 MQRC-CRYPTO-HARDWARE-ERROR      PIC S9(9) BINARY VALUE 2382.                          1636
COPIED          10 MQRC-AUTH-INFO-REC-COUNT-ERROR  PIC S9(9) BINARY VALUE 2383.                          1637
COPIED          10 MQRC-AUTH-INFO-REC-ERROR        PIC S9(9) BINARY VALUE 2384.                          1638
COPIED          10 MQRC-AIR-ERROR                  PIC S9(9) BINARY VALUE 2385.                          1639
COPIED          10 MQRC-AUTH-INFO-TYPE-ERROR       PIC S9(9) BINARY VALUE 2386.                          1640
COPIED          10 MQRC-AUTH-INFO-CONN-NAME-ERROR  PIC S9(9) BINARY VALUE 2387.                          1641
COPIED          10 MQRC-LDAP-USER-NAME-ERROR       PIC S9(9) BINARY VALUE 2388.                          1642
COPIED          10 MQRC-LDAP-USER-NAME-LENGTH-ERR  PIC S9(9) BINARY VALUE 2389.                          1643
COPIED          10 MQRC-LDAP-PASSWORD-ERROR        PIC S9(9) BINARY VALUE 2390.                          1644
COPIED          10 MQRC-SSL-ALREADY-INITIALIZED    PIC S9(9) BINARY VALUE 2391.                          1645
COPIED          10 MQRC-SSL-CONFIG-ERROR           PIC S9(9) BINARY VALUE 2392.                          1646
COPIED          10 MQRC-SSL-INITIALIZATION-ERROR   PIC S9(9) BINARY VALUE 2393.                          1647
COPIED          10 MQRC-Q-INDEX-TYPE-ERROR         PIC S9(9) BINARY VALUE 2394.                          1648
COPIED          10 MQRC-SSL-NOT-ALLOWED            PIC S9(9) BINARY VALUE 2396.                          1649
COPIED          10 MQRC-JSSE-ERROR                 PIC S9(9) BINARY VALUE 2397.                          1650
COPIED          10 MQRC-SSL-PEER-NAME-MISMATCH     PIC S9(9) BINARY VALUE 2398.                          1651
COPIED          10 MQRC-SSL-PEER-NAME-ERROR        PIC S9(9) BINARY VALUE 2399.                          1652
COPIED          10 MQRC-UNSUPPORTED-CIPHER-SUITE   PIC S9(9) BINARY VALUE 2400.                          1653
COPIED          10 MQRC-SSL-CERTIFICATE-REVOKED    PIC S9(9) BINARY VALUE 2401.                          1654
COPIED          10 MQRC-SSL-CERT-STORE-ERROR       PIC S9(9) BINARY VALUE 2402.                          1655
COPIED          10 MQRC-REOPEN-EXCL-INPUT-ERROR    PIC S9(9) BINARY VALUE 6100.                          1656
COPIED          10 MQRC-REOPEN-INQUIRE-ERROR       PIC S9(9) BINARY VALUE 6101.                          1657
COPIED          10 MQRC-REOPEN-SAVED-CONTEXT-ERR   PIC S9(9) BINARY VALUE 6102.                          1658
COPIED          10 MQRC-REOPEN-TEMPORARY-Q-ERROR   PIC S9(9) BINARY VALUE 6103.                          1659
COPIED          10 MQRC-ATTRIBUTE-LOCKED           PIC S9(9) BINARY VALUE 6104.                          1660
COPIED          10 MQRC-CURSOR-NOT-VALID           PIC S9(9) BINARY VALUE 6105.                          1661
COPIED          10 MQRC-ENCODING-ERROR             PIC S9(9) BINARY VALUE 6106.                          1662
COPIED          10 MQRC-STRUC-ID-ERROR             PIC S9(9) BINARY VALUE 6107.                          1663
COPIED          10 MQRC-NULL-POINTER               PIC S9(9) BINARY VALUE 6108.                          1664
COPIED          10 MQRC-NO-CONNECTION-REFERENCE    PIC S9(9) BINARY VALUE 6109.                          1665
COPIED          10 MQRC-NO-BUFFER                  PIC S9(9) BINARY VALUE 6110.                          1666

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    35
       1--------------------VISION:RESULTS FREE FORM TEXT----------------------72         --------
COPIED          10 MQRC-BINARY-DATA-LENGTH-ERROR   PIC S9(9) BINARY VALUE 6111.                          1667
COPIED          10 MQRC-BUFFER-NOT-AUTOMATIC       PIC S9(9) BINARY VALUE 6112.                          1668
COPIED          10 MQRC-INSUFFICIENT-BUFFER        PIC S9(9) BINARY VALUE 6113.                          1669
COPIED          10 MQRC-INSUFFICIENT-DATA          PIC S9(9) BINARY VALUE 6114.                          1670
```

Figure 22 MQGET Sample Compiled Listing (Page 21 of 35)

```
COPIED        10 MQRC-DATA-TRUNCATED            PIC S9(9) BINARY VALUE 6115.                    1671
COPIED        10 MQRC-ZERO-LENGTH               PIC S9(9) BINARY VALUE 6116.                    1672
COPIED        10 MQRC-NEGATIVE-LENGTH           PIC S9(9) BINARY VALUE 6117.                    1673
COPIED        10 MQRC-NEGATIVE-OFFSET           PIC S9(9) BINARY VALUE 6118.                    1674
COPIED        10 MQRC-INCONSISTENT-FORMAT       PIC S9(9) BINARY VALUE 6119.                    1675
COPIED        10 MQRC-INCONSISTENT-OBJECT-STATE PIC S9(9) BINARY VALUE 6120.                    1676
COPIED        10 MQRC-CONTEXT-OBJECT-NOT-VALID  PIC S9(9) BINARY VALUE 6121.                    1677
COPIED        10 MQRC-CONTEXT-OPEN-ERROR        PIC S9(9) BINARY VALUE 6122.                    1678
COPIED        10 MQRC-STRUC-LENGTH-ERROR        PIC S9(9) BINARY VALUE 6123.                    1679
COPIED        10 MQRC-NOT-CONNECTED             PIC S9(9) BINARY VALUE 6124.                    1680
COPIED        10 MQRC-NOT-OPEN                  PIC S9(9) BINARY VALUE 6125.                    1681
COPIED        10 MQRC-DISTRIBUTION-LIST-EMPTY   PIC S9(9) BINARY VALUE 6126.                    1682
COPIED        10 MQRC-INCONSISTENT-OPEN-OPTIONS PIC S9(9) BINARY VALUE 6127.                    1683
COPIED        10 MQRC-WRONG-VERSION             PIC S9(9) BINARY VALUE 6128.                    1684
COPIED        10 MQRC-REFERENCE-ERROR           PIC S9(9) BINARY VALUE 6129.                    1685
COPIED                                                                                          1686
COPIED                                                                                          1687
COPIED      ******************************************************************                  1688
COPIED      **  Values Related to Queue Attributes                          **                  1689
COPIED      ******************************************************************                  1690
COPIED                                                                                          1691
COPIED      **   Queue Types                                                                    1692
COPIED        10 MQQT-LOCAL   PIC S9(9) BINARY VALUE 1.                                        1693
COPIED        10 MQQT-MODEL   PIC S9(9) BINARY VALUE 2.                                        1694
COPIED        10 MQQT-ALIAS   PIC S9(9) BINARY VALUE 3.                                        1695
COPIED        10 MQQT-REMOTE  PIC S9(9) BINARY VALUE 6.                                        1696
COPIED        10 MQQT-CLUSTER PIC S9(9) BINARY VALUE 7.                                        1697
COPIED                                                                                          1698
COPIED      **   Cluster Queue Types                                                            1699
COPIED        10 MQCQT-LOCAL-Q     PIC S9(9) BINARY VALUE 1.                                   1700
COPIED        10 MQCQT-ALIAS-Q     PIC S9(9) BINARY VALUE 2.                                   1701
COPIED        10 MQCQT-REMOTE-Q    PIC S9(9) BINARY VALUE 3.                                   1702
COPIED        10 MQCQT-Q-MGR-ALIAS PIC S9(9) BINARY VALUE 4.                                   1703
COPIED                                                                                          1704
COPIED      **   Extended Queue Types                                                           1705
COPIED        10 MQQT-ALL PIC S9(9) BINARY VALUE 1001.                                         1706
COPIED                                                                                          1707
COPIED      **   Queue Definition Types                                                         1708
COPIED        10 MQQDT-PREDEFINED        PIC S9(9) BINARY VALUE 1.                            1709
COPIED        10 MQQDT-PERMANENT-DYNAMIC PIC S9(9) BINARY VALUE 2.                            1710
COPIED        10 MQQDT-TEMPORARY-DYNAMIC PIC S9(9) BINARY VALUE 3.                            1711
COPIED        10 MQQDT-SHARED-DYNAMIC    PIC S9(9) BINARY VALUE 4.                            1712
COPIED                                                                                          1713
COPIED      **   Inhibit Get                                                                    1714
COPIED        10 MQQA-GET-INHIBITED PIC S9(9) BINARY VALUE 1.                                  1715

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    36
      1--------------------VISION:RESULTS FREE FORM TEXT----------------------72         --------
COPIED        10 MQQA-GET-ALLOWED   PIC S9(9) BINARY VALUE 0.                                  1716
COPIED                                                                                          1717
COPIED      **   Inhibit Put                                                                    1718
COPIED        10 MQQA-PUT-INHIBITED PIC S9(9) BINARY VALUE 1.                                  1719
COPIED        10 MQQA-PUT-ALLOWED   PIC S9(9) BINARY VALUE 0.                                  1720
COPIED                                                                                          1721
COPIED      **   Queue Shareability                                                             1722
COPIED        10 MQQA-SHAREABLE     PIC S9(9) BINARY VALUE 1.                                  1723
COPIED        10 MQQA-NOT-SHAREABLE PIC S9(9) BINARY VALUE 0.                                  1724
COPIED                                                                                          1725
COPIED      **   Back-Out Hardening                                                             1726
COPIED        10 MQQA-BACKOUT-HARDENED     PIC S9(9) BINARY VALUE 1.                       1727
COPIED        10 MQQA-BACKOUT-NOT-HARDENED PIC S9(9) BINARY VALUE 0.                          1728
COPIED                                                                                          1729
COPIED      **   Message Delivery Sequence                                                      1730
COPIED        10 MQMDS-PRIORITY PIC S9(9) BINARY VALUE 0.                                      1731
COPIED        10 MQMDS-FIFO     PIC S9(9) BINARY VALUE 1.                                      1732
COPIED                                                                                          1733
COPIED      **   Trigger Control                                                                1734
COPIED        10 MQTC-OFF PIC S9(9) BINARY VALUE 0.                                            1735
COPIED        10 MQTC-ON  PIC S9(9) BINARY VALUE 1.                                            1736
COPIED                                                                                          1737
COPIED      **   Trigger Types                                                                  1738
COPIED        10 MQTT-NONE  PIC S9(9) BINARY VALUE 0.                                          1739
COPIED        10 MQTT-FIRST PIC S9(9) BINARY VALUE 1.                                          1740
COPIED        10 MQTT-EVERY PIC S9(9) BINARY VALUE 2.                                          1741
COPIED        10 MQTT-DEPTH PIC S9(9) BINARY VALUE 3.                                          1742
COPIED                                                                                          1743
COPIED      **   Queue Usage                                                                    1744
COPIED        10 MQUS-NORMAL       PIC S9(9) BINARY VALUE 0.                                   1745
COPIED        10 MQUS-TRANSMISSION PIC S9(9) BINARY VALUE 1.                                   1746
COPIED                                                                                          1747
COPIED      **   Distribution Lists                                                             1748
COPIED        10 MQDL-SUPPORTED     PIC S9(9) BINARY VALUE 1.                                  1749
COPIED        10 MQDL-NOT-SUPPORTED PIC S9(9) BINARY VALUE 0.                                  1750
COPIED                                                                                          1751
COPIED      **   Index Type                                                                     1752
```

Figure 22 MQGET Sample Compiled Listing (Page 22 of 35)

```
COPIED        10 MQIT-NONE      PIC S9(9) BINARY VALUE 0.                                1753
COPIED        10 MQIT-MSG-ID    PIC S9(9) BINARY VALUE 1.                                1754
COPIED        10 MQIT-CORREL-ID PIC S9(9) BINARY VALUE 2.                                1755
COPIED        10 MQIT-MSG-TOKEN PIC S9(9) BINARY VALUE 4.                                1756
COPIED        10 MQIT-GROUP-ID  PIC S9(9) BINARY VALUE 5.                                1757
COPIED                                                                                   1758
COPIED      **   Default Bind                                                            1759
COPIED        10 MQBND-BIND-ON-OPEN   PIC S9(9) BINARY VALUE 0.                          1760
COPIED        10 MQBND-BIND-NOT-FIXED PIC S9(9) BINARY VALUE 1.                          1761
COPIED                                                                                   1762
COPIED      **   Queue Sharing Group Disposition                                         1763
COPIED        10 MQQSGD-Q-MGR  PIC S9(9) BINARY VALUE 0.                                 1764

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    37
      1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED        10 MQQSGD-COPY   PIC S9(9) BINARY VALUE 1.                                1765
COPIED        10 MQQSGD-SHARED PIC S9(9) BINARY VALUE 2.                                1766
COPIED        10 MQQSGD-GROUP  PIC S9(9) BINARY VALUE 3.                                1767
COPIED                                                                                   1768
COPIED                                                                                   1769
COPIED      ************************************************************************     1770
COPIED      **  Values Related to Namelist Attributes                             **     1771
COPIED      ************************************************************************     1772
COPIED                                                                                   1773
COPIED      **   Name Count                                                              1774
COPIED        10 MQNC-MAX-NAMELIST-NAME-COUNT PIC S9(9) BINARY VALUE 256.              1775
COPIED                                                                                   1776
COPIED      **   Namelist Type                                                           1777
COPIED        10 MQNT-NONE      PIC S9(9) BINARY VALUE 0.                                1778
COPIED        10 MQNT-Q         PIC S9(9) BINARY VALUE 1.                                1779
COPIED        10 MQNT-CLUSTER   PIC S9(9) BINARY VALUE 2.                                1780
COPIED        10 MQNT-AUTH-INFO PIC S9(9) BINARY VALUE 4.                                1781
COPIED        10 MQNT-ALL       PIC S9(9) BINARY VALUE 1001.                             1782
COPIED                                                                                   1783
COPIED                                                                                   1784
COPIED      ************************************************************************     1785
COPIED      **  Values Related to Process-Definition Attributes                   **     1786
COPIED      ************************************************************************     1787
COPIED                                                                                   1788
COPIED      **   Application Type                                                        1789
COPIED      **   See values for "Put Application Type" under MQMD                        1790
COPIED                                                                                   1791
COPIED                                                                                   1792
COPIED      ************************************************************************     1793
COPIED      **  Values Related to Authentication-Information Attributes           **     1794
COPIED      ************************************************************************     1795
COPIED                                                                                   1796
COPIED      **   Authentication Information Type                                         1797
COPIED      **   See values for "Authentication Information Type" under MQAIR            1798
COPIED                                                                                   1799
COPIED                                                                                   1800
COPIED      ************************************************************************     1801
COPIED      **  Values Related to Queue-Manager Attributes                        **     1802
COPIED      ************************************************************************     1803
COPIED                                                                                   1804
COPIED      **   Channel Auto Definition                                                 1805
COPIED        10 MQCHAD-DISABLED PIC S9(9) BINARY VALUE 0.                             1806
COPIED        10 MQCHAD-ENABLED  PIC S9(9) BINARY VALUE 1.                             1807
COPIED                                                                                   1808
COPIED      **   Command Level                                                           1809
COPIED        10 MQCMDL-LEVEL-1   PIC S9(9) BINARY VALUE 100.                          1810
COPIED        10 MQCMDL-LEVEL-101 PIC S9(9) BINARY VALUE 101.                          1811
COPIED        10 MQCMDL-LEVEL-110 PIC S9(9) BINARY VALUE 110.                          1812
COPIED        10 MQCMDL-LEVEL-114 PIC S9(9) BINARY VALUE 114.                          1813

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    38
      1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED        10 MQCMDL-LEVEL-120 PIC S9(9) BINARY VALUE 120.                          1814
COPIED        10 MQCMDL-LEVEL-200 PIC S9(9) BINARY VALUE 200.                          1815
COPIED        10 MQCMDL-LEVEL-201 PIC S9(9) BINARY VALUE 201.                          1816
COPIED        10 MQCMDL-LEVEL-210 PIC S9(9) BINARY VALUE 210.                          1817
COPIED        10 MQCMDL-LEVEL-220 PIC S9(9) BINARY VALUE 220.                          1818
COPIED        10 MQCMDL-LEVEL-221 PIC S9(9) BINARY VALUE 221.                          1819
COPIED        10 MQCMDL-LEVEL-320 PIC S9(9) BINARY VALUE 320.                          1820
COPIED        10 MQCMDL-LEVEL-420 PIC S9(9) BINARY VALUE 420.                          1821
COPIED        10 MQCMDL-LEVEL-500 PIC S9(9) BINARY VALUE 500.                          1822
COPIED        10 MQCMDL-LEVEL-510 PIC S9(9) BINARY VALUE 510.                          1823
COPIED        10 MQCMDL-LEVEL-520 PIC S9(9) BINARY VALUE 520.                          1824
COPIED        10 MQCMDL-LEVEL-530 PIC S9(9) BINARY VALUE 530.                          1825
COPIED                                                                                   1826
COPIED      **   Distribution Lists                                                      1827
COPIED      **   See values for "Distribution Lists" under Queue Attributes              1828
COPIED                                                                                   1829
COPIED      **   Expiration Scan Interval                                                1830
COPIED        10 MQEXPI-OFF PIC S9(9) BINARY VALUE 0.                                  1831
```

Figure 22 MQGET Sample Compiled Listing (Page 23 of 35)

```
   COPIED
   COPIED      **  Intra-Group Queuing
   COPIED        10 MQIGQ-DISABLED PIC S9(9) BINARY VALUE 0.
   COPIED        10 MQIGQ-ENABLED  PIC S9(9) BINARY VALUE 1.
   COPIED
   COPIED      **  Intra-Group Queuing Put Authority
   COPIED        10 MQIGQPA-DEFAULT          PIC S9(9) BINARY VALUE 1.
   COPIED        10 MQIGQPA-CONTEXT          PIC S9(9) BINARY VALUE 2.
   COPIED        10 MQIGQPA-ONLY-IGQ         PIC S9(9) BINARY VALUE 3.
   COPIED        10 MQIGQPA-ALTERNATE-OR-IGQ PIC S9(9) BINARY VALUE 4.
   COPIED
   COPIED      **  Platform
   COPIED        10 MQPL-MVS        PIC S9(9) BINARY VALUE 1.
   COPIED        10 MQPL-OS390      PIC S9(9) BINARY VALUE 1.
   COPIED        10 MQPL-ZOS        PIC S9(9) BINARY VALUE 1.
   COPIED        10 MQPL-OS2        PIC S9(9) BINARY VALUE 2.
   COPIED        10 MQPL-AIX        PIC S9(9) BINARY VALUE 3.
   COPIED        10 MQPL-UNIX       PIC S9(9) BINARY VALUE 3.
   COPIED        10 MQPL-OS400      PIC S9(9) BINARY VALUE 4.
   COPIED        10 MQPL-WINDOWS    PIC S9(9) BINARY VALUE 5.
   COPIED        10 MQPL-WINDOWS-NT PIC S9(9) BINARY VALUE 11.
   COPIED        10 MQPL-VMS        PIC S9(9) BINARY VALUE 12.
   COPIED        10 MQPL-NSK        PIC S9(9) BINARY VALUE 13.
   COPIED        10 MQPL-NATIVE     PIC S9(9) BINARY VALUE 1.
   COPIED
   COPIED      **  Syncpoint Availability
   COPIED        10 MQSP-AVAILABLE     PIC S9(9) BINARY VALUE 1.
   COPIED        10 MQSP-NOT-AVAILABLE PIC S9(9) BINARY VALUE 0.
WARNING   DYL-187W   ZERO LENGTH DATANAME OF  CHARATTRS  MUST NOT BE REFERENCED
   COPIED
   COPIED

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE   39
       1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
   COPIED      ********************************************************************
   COPIED      **  End of CMQV                                                   **
   COPIED      ********************************************************************
       WORKAREA
       COPY CMQCNOV COBOL
   COPIED      ********************************************************************
   COPIED      **                                                                **
   COPIED      **                  WebSphere MQ for z/OS                         **
   COPIED      **                                                                **
   COPIED      **  FILE NAME:      CMQCNOV                                       **
   COPIED      **                                                                **
   COPIED      **  DESCRIPTION:    Connect Options Structure                     **
   COPIED      **                                                                **
   COPIED      ********************************************************************
   COPIED      **  @START_COPYRIGHT@                                             **
   COPIED      **  Statement:     Licensed Materials - Property of IBM           **
   COPIED      **                                                                **
   COPIED      **                 5655-F10                                       **
   COPIED      **                 (C) Copyright IBM Corporation. 1997, 2002      **
   COPIED      **                                                                **
   COPIED      **  Status:        Version 5 Release 3                            **
   COPIED      **  @END_COPYRIGHT@                                               **
   COPIED      ********************************************************************
   COPIED      **                                                                **
   COPIED      **  FUNCTION:       This file declares the structure MQCNO,       **
   COPIED      **                  which is used by the main MQI.                **
   COPIED      **                                                                **
   COPIED      **  PROCESSOR:      COBOL                                         **
   COPIED      **                                                                **
   COPIED      ********************************************************************
   COPIED
   COPIED      **  MQCNO structure
   COPIED        10 MQCNO.
   COPIED      **   Structure identifier
   COPIED         15 MQCNO-STRUCID          PIC X(4) VALUE 'CNO '.
   COPIED      **   Structure version number
   COPIED         15 MQCNO-VERSION          PIC S9(9) BINARY VALUE 1.
   COPIED      **   Options that control the action of MQCONNX
   COPIED         15 MQCNO-OPTIONS          PIC S9(9) BINARY VALUE 0.
   COPIED      **   Offset of MQCD structure for client connection
   COPIED         15 MQCNO-CLIENTCONNOFFSET PIC S9(9) BINARY VALUE 0.
   COPIED      **   Address of MQCD structure for client connection
   COPIED         15 MQCNO-CLIENTCONNPTR    POINTER VALUE NULL.
   COPIED      **   Queue-manager connection tag
   COPIED         15 MQCNO-CONNTAG          PIC X(128) VALUE LOW-VALUES.
   COPIED
   COPIED      ********************************************************************
   COPIED      **  End of CMQCNOV                                                **
   COPIED      ********************************************************************

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE   40
```

Figure 22   MQGET Sample Compiled Listing (Page 24 of 35)

```
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
      WORKAREA                                                                                1911
      COPY CMQMDV COBOL                                                                       1912
COPIED      ************************************************************************        1913
COPIED      **                                                                    **        1914
COPIED      **                 WebSphere MQ for z/OS                              **        1915
COPIED      **                                                                    **        1916
COPIED      **  FILE NAME:      CMQMDV                                            **        1917
COPIED      **                                                                    **        1918
COPIED      **  DESCRIPTION:    Message Descriptor Structure                      **        1919
COPIED      **                                                                    **        1920
COPIED      ************************************************************************        1921
COPIED      **  @START_COPYRIGHT@                                                 **        1922
COPIED      **  Statement:     Licensed Materials - Property of IBM               **        1923
COPIED      **                                                                    **        1924
COPIED      **                 5655-F10                                           **        1925
COPIED      **                 (C) Copyright IBM Corporation. 1993, 2002          **        1926
COPIED      **                                                                    **        1927
COPIED      **  Status:        Version 5 Release 3                               **        1928
COPIED      **  @END_COPYRIGHT@                                                   **        1929
COPIED      ************************************************************************        1930
COPIED      **                                                                    **        1931
COPIED      **  FUNCTION:       This file declares the structure MQMD,            **        1932
COPIED      **                  which is used by the main MQI.                    **        1933
COPIED      **                                                                    **        1934
COPIED      **  PROCESSOR:      COBOL                                             **        1935
COPIED      **                                                                    **        1936
COPIED      ************************************************************************        1937
COPIED                                                                                      1938
COPIED      **   MQMD structure                                                             1939
COPIED        10 MQMD.                                                                      1940
COPIED      **    Structure identifier                                                      1941
COPIED         15 MQMD-STRUCID          PIC X(4) VALUE 'MD  '.                          1942
COPIED      **    Structure version number                                                  1943
COPIED         15 MQMD-VERSION          PIC S9(9) BINARY VALUE 1.                       1944
COPIED      **    Options for report messages                                               1945
COPIED         15 MQMD-REPORT           PIC S9(9) BINARY VALUE 0.                       1946
COPIED      **    Message type                                                              1947
COPIED         15 MQMD-MSGTYPE          PIC S9(9) BINARY VALUE 8.                       1948
COPIED      **    Message lifetime                                                          1949
COPIED         15 MQMD-EXPIRY           PIC S9(9) BINARY VALUE -1.                      1950
COPIED      **    Feedback or reason code                                                   1951
COPIED         15 MQMD-FEEDBACK         PIC S9(9) BINARY VALUE 0.                       1952
COPIED      **    Numeric encoding of message data                                          1953
COPIED         15 MQMD-ENCODING         PIC S9(9) BINARY VALUE 785.                     1954
COPIED      **    Character set identifier of message data                                  1955
COPIED         15 MQMD-CODEDCHARSETID   PIC S9(9) BINARY VALUE 0.                       1956
COPIED      **    Format name of message data                                               1957
COPIED         15 MQMD-FORMAT           PIC X(8) VALUE SPACES.                          1958
COPIED      **    Message priority                                                          1959

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    41
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
COPIED         15 MQMD-PRIORITY         PIC S9(9) BINARY VALUE -1.                      1960
COPIED      **    Message persistence                                                       1961
COPIED         15 MQMD-PERSISTENCE      PIC S9(9) BINARY VALUE 2.                       1962
COPIED      **    Message identifier                                                        1963
COPIED         15 MQMD-MSGID            PIC X(24) VALUE LOW-VALUES.                     1964
COPIED      **    Correlation identifier                                                    1965
COPIED         15 MQMD-CORRELID         PIC X(24) VALUE LOW-VALUES.                     1966
COPIED      **    Backout counter                                                           1967
COPIED         15 MQMD-BACKOUTCOUNT     PIC S9(9) BINARY VALUE 0.                       1968
COPIED      **    Name of reply queue                                                       1969
COPIED         15 MQMD-REPLYTOQ         PIC X(48) VALUE SPACES.                         1970
COPIED      **    Name of reply queue manager                                               1971
COPIED         15 MQMD-REPLYTOQMGR      PIC X(48) VALUE SPACES.                         1972
COPIED      **    User identifier                                                           1973
COPIED         15 MQMD-USERIDENTIFIER   PIC X(12) VALUE SPACES.                         1974
COPIED      **    Accounting token                                                          1975
COPIED         15 MQMD-ACCOUNTINGTOKEN  PIC X(32) VALUE LOW-VALUES.                     1976
COPIED      **    Application data relating to identity                                     1977
COPIED         15 MQMD-APPLIDENTITYDATA PIC X(32) VALUE SPACES.                         1978
COPIED      **    Type of application that put the message                                  1979
COPIED         15 MQMD-PUTAPPLTYPE      PIC S9(9) BINARY VALUE 0.                       1980
COPIED      **    Name of application that put the message                                  1981
COPIED         15 MQMD-PUTAPPLNAME      PIC X(28) VALUE SPACES.                         1982
COPIED      **    Date when message was put                                                 1983
COPIED         15 MQMD-PUTDATE          PIC X(8) VALUE SPACES.                          1984
COPIED      **    Time when message was put                                                 1985
COPIED         15 MQMD-PUTTIME          PIC X(8) VALUE SPACES.                          1986
COPIED      **    Application data relating to origin                                       1987
COPIED         15 MQMD-APPLORIGINDATA   PIC X(4) VALUE SPACES.                          1988
COPIED      **    Group identifier                                                          1989
COPIED         15 MQMD-GROUPID          PIC X(24) VALUE LOW-VALUES.                     1990
COPIED      **    Sequence number of logical message within group                           1991
```

Figure 22 MQGET Sample Compiled Listing (Page 25 of 35)

```
COPIED         15 MQMD-MSGSEQNUMBER     PIC S9(9) BINARY VALUE 1.                              1992
COPIED       **   Offset of data in physical message from start of logical                     1993
COPIED       **   message                                                                      1994
COPIED         15 MQMD-OFFSET           PIC S9(9) BINARY VALUE 0.                              1995
COPIED       **   Message flags                                                                1996
COPIED         15 MQMD-MSGFLAGS         PIC S9(9) BINARY VALUE 0.                              1997
COPIED       **   Length of original message                                                   1998
COPIED         15 MQMD-ORIGINALLENGTH   PIC S9(9) BINARY VALUE -1.                             1999
COPIED                                                                                         2000
COPIED       ******************************************************************              2001
COPIED       **  End of CMQMDV                                               **              2002
COPIED       ******************************************************************              2003
       WORKAREA                                                                                2004
       COPY CMQGMOV COBOL                                                                      2005
COPIED       ******************************************************************              2006
COPIED       **                                                              **              2007
COPIED       **                WebSphere MQ for z/OS                         **              2008

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    42
       1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED       **                                                              **              2009
COPIED       **  FILE NAME:      CMQGMOV                                     **              2010
COPIED       **                                                              **              2011
COPIED       **  DESCRIPTION:    Get Message Options Structure               **              2012
COPIED       **                                                              **              2013
COPIED       ******************************************************************              2014
COPIED       **  @START_COPYRIGHT@                                           **              2015
COPIED       **  Statement:     Licensed Materials - Property of IBM         **              2016
COPIED       **                                                              **              2017
COPIED       **                 5655-F10                                     **              2018
COPIED       **                 (C) Copyright IBM Corporation. 1993, 2002    **              2019
COPIED       **                                                              **              2020
COPIED       **  Status:        Version 5 Release 3                          **              2021
COPIED       **  @END_COPYRIGHT@                                             **              2022
COPIED       ******************************************************************              2023
COPIED       **                                                              **              2024
COPIED       **  FUNCTION:       This file declares the structure MQGMO,     **              2025
COPIED       **                  which is used by the main MQI.              **              2026
COPIED       **                                                              **              2027
COPIED       **  PROCESSOR:      COBOL                                       **              2028
COPIED       **                                                              **              2029
COPIED       ******************************************************************              2030
COPIED                                                                                         2031
COPIED       **  MQGMO structure                                                             2032
COPIED         10 MQGMO.                                                                       2033
COPIED       **   Structure identifier                                                        2034
COPIED         15 MQGMO-STRUCID        PIC X(4) VALUE 'GMO '.                                 2035
COPIED       **   Structure version number                                                    2036
COPIED         15 MQGMO-VERSION        PIC S9(9) BINARY VALUE 1.                              2037
COPIED       **   Options that control the action of MQGET                                    2038
COPIED         15 MQGMO-OPTIONS        PIC S9(9) BINARY VALUE 0.                              2039
COPIED       **   Wait interval                                                               2040
COPIED         15 MQGMO-WAITINTERVAL   PIC S9(9) BINARY VALUE 0.                              2041
COPIED       **   Pointer to signal                                                           2042
COPIED         15 MQGMO-SIGNAL1        POINTER VALUE NULL.                                    2043
COPIED       **   Signal identifier                                                           2044
COPIED         15 MQGMO-SIGNAL2        PIC S9(9) BINARY VALUE 0.                              2045
COPIED       **   Resolved name of destination queue                                          2046
COPIED         15 MQGMO-RESOLVEDQNAME  PIC X(48) VALUE SPACES.                                2047
COPIED       **   Options controlling selection criteria used for MQGET                       2048
COPIED         15 MQGMO-MATCHOPTIONS   PIC S9(9) BINARY VALUE 3.                              2049
COPIED       **   Flag indicating whether message retrieved is in a group                     2050
COPIED         15 MQGMO-GROUPSTATUS    PIC X VALUE ' '.                                       2051
COPIED       **   Flag indicating whether message retrieved is a segment of a                 2052
COPIED       **   logical message                                                             2053
COPIED         15 MQGMO-SEGMENTSTATUS  PIC X VALUE ' '.                                       2054
COPIED       **   Flag indicating whether further segmentation is allowed for                 2055
COPIED       **   the message retrieved                                                       2056
COPIED         15 MQGMO-SEGMENTATION   PIC X VALUE ' '.                                       2057

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    43
       1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED       **   Reserved                                                                    2058
COPIED         15 MQGMO-RESERVED1      PIC X VALUE SPACES.                                    2059
COPIED       **   Message token                                                               2060
COPIED         15 MQGMO-MSGTOKEN       PIC X(16) VALUE LOW-VALUES.                            2061
COPIED       **   Length of message data returned (bytes)                                     2062
COPIED         15 MQGMO-RETURNEDLENGTH PIC S9(9) BINARY VALUE -1.                             2063
COPIED                                                                                         2064
COPIED       ******************************************************************              2065
COPIED       **  End of CMQGMOV                                              **              2066
COPIED       ******************************************************************              2067
       WORKAREA                                                                                2068
       COPY CMQODV COBOL                                                                       2069
COPIED       ******************************************************************              2070
```

Figure 22 MQGET Sample Compiled Listing (Page 26 of 35)

```
COPIED      **                                                              **  2071
COPIED      **                 WebSphere MQ for z/OS                        **  2072
COPIED      **                                                              **  2073
COPIED      **  FILE NAME:      CMQODV                                      **  2074
COPIED      **                                                              **  2075
COPIED      **  DESCRIPTION:    Object Descriptor Structure                 **  2076
COPIED      **                                                              **  2077
COPIED      ******************************************************************  2078
COPIED      **  @START_COPYRIGHT@                                           **  2079
COPIED      **  Statement:      Licensed Materials - Property of IBM        **  2080
COPIED      **                                                              **  2081
COPIED      **                 5655-F10                                     **  2082
COPIED      **                 (C) Copyright IBM Corporation. 1993, 2002    **  2083
COPIED      **                                                              **  2084
COPIED      **  Status:         Version 5 Release 3                         **  2085
COPIED      **  @END_COPYRIGHT@                                             **  2086
COPIED      ******************************************************************  2087
COPIED      **                                                              **  2088
COPIED      **  FUNCTION:       This file declares the structure MQOD,      **  2089
COPIED      **                  which is used by the main MQI.              **  2090
COPIED      **                                                              **  2091
COPIED      **  PROCESSOR:      COBOL                                       **  2092
COPIED      **                                                              **  2093
COPIED      ******************************************************************  2094
COPIED                                                                          2095
COPIED      **   MQOD structure                                                 2096
COPIED        10 MQOD.                                                          2097
COPIED      **    Structure identifier                                        2098
COPIED          15 MQOD-STRUCID            PIC X(4) VALUE 'OD  '.           2099
COPIED      **    Structure version number                                    2100
COPIED          15 MQOD-VERSION            PIC S9(9) BINARY VALUE 1.        2101
COPIED      **    Object type                                                 2102
COPIED          15 MQOD-OBJECTTYPE         PIC S9(9) BINARY VALUE 1.        2103
COPIED      **    Object name                                                 2104
COPIED          15 MQOD-OBJECTNAME         PIC X(48) VALUE SPACES.          2105
COPIED      **    Object queue manager name                                   2106

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    44
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED          15 MQOD-OBJECTQMGRNAME     PIC X(48) VALUE SPACES.          2107
COPIED      **    Dynamic queue name                                          2108
COPIED          15 MQOD-DYNAMICQNAME       PIC X(48) VALUE 'CSQ.*'.         2109
COPIED      **    Alternate user identifier                                   2110
COPIED          15 MQOD-ALTERNATEUSERID    PIC X(12) VALUE SPACES.          2111
COPIED      **    Number of object records present                            2112
COPIED          15 MQOD-RECSPRESENT        PIC S9(9) BINARY VALUE 0.        2113
COPIED      **    Number of local queues opened successfully                  2114
COPIED          15 MQOD-KNOWNDESTCOUNT     PIC S9(9) BINARY VALUE 0.        2115
COPIED      **    Number of remote queues opened successfully                 2116
COPIED          15 MQOD-UNKNOWNDESTCOUNT   PIC S9(9) BINARY VALUE 0.        2117
COPIED      **    Number of queues that failed to open                        2118
COPIED          15 MQOD-INVALIDDESTCOUNT   PIC S9(9) BINARY VALUE 0.        2119
COPIED      **    Offset of first object record from start of MQOD            2120
COPIED          15 MQOD-OBJECTRECOFFSET    PIC S9(9) BINARY VALUE 0.        2121
COPIED      **    Offset of first response record from start of MQOD          2122
COPIED          15 MQOD-RESPONSERECOFFSET  PIC S9(9) BINARY VALUE 0.        2123
COPIED      **    Address of first object record                              2124
COPIED          15 MQOD-OBJECTRECPTR       POINTER VALUE NULL.              2125
COPIED      **    Address of first response record                            2126
COPIED          15 MQOD-RESPONSERECPTR     POINTER VALUE NULL.              2127
COPIED      **    Alternate security identifier                               2128
COPIED          15 MQOD-ALTERNATESECURITYID PIC X(40) VALUE LOW-VALUES.      2129
COPIED      **    Resolved queue name                                         2130
COPIED          15 MQOD-RESOLVEDQNAME      PIC X(48) VALUE SPACES.          2131
COPIED      **    Resolved queue manager name                                 2132
COPIED          15 MQOD-RESOLVEDQMGRNAME   PIC X(48) VALUE SPACES.          2133
COPIED                                                                          2134
COPIED      ******************************************************************  2135
COPIED      **  End of CMQODV                                               **  2136
COPIED      ******************************************************************  2137
      WORKAREA                                                                  2138
      COPY CMQPMOV COBOL                                                        2139
COPIED      ******************************************************************  2140
COPIED      **                                                              **  2141
COPIED      **                 WebSphere MQ for z/OS                        **  2142
COPIED      **                                                              **  2143
COPIED      **  FILE NAME:      CMQPMOV                                     **  2144
COPIED      **                                                              **  2145
COPIED      **  DESCRIPTION:    Put Message Options Structure               **  2146
COPIED      **                                                              **  2147
COPIED      ******************************************************************  2148
COPIED      **  @START_COPYRIGHT@                                           **  2149
COPIED      **  Statement:      Licensed Materials - Property of IBM        **  2150
COPIED      **                                                              **  2151
COPIED      **                 5655-F10                                     **  2152
```

Figure 22 MQGET Sample Compiled Listing (Page 27 of 35)

```
COPIED      **               (C) Copyright IBM Corporation. 1993, 2002    **                              2153
COPIED      **                                                            **                              2154
COPIED      **  Status:       Version 5 Release 3                         **                              2155

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    45
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED      **  @END_COPYRIGHT@                                           **                              2156
COPIED      ****************************************************************                              2157
COPIED      **                                                            **                              2158
COPIED      **  FUNCTION:       This file declares the structure MQPMO,   **                              2159
COPIED      **                  which is used by the main MQI.            **                              2160
COPIED      **                                                            **                              2161
COPIED      **  PROCESSOR:      COBOL                                     **                              2162
COPIED      **                                                            **                              2163
COPIED      ****************************************************************                              2164
COPIED                                                                                                    2165
COPIED      **   MQPMO structure                                                                          2166
COPIED        10 MQPMO.                                                                                 2167
COPIED      **    Structure identifier                                                                    2168
COPIED          15 MQPMO-STRUCID          PIC X(4) VALUE 'PMO '.                                        2169
COPIED      **    Structure version number                                                                2170
COPIED          15 MQPMO-VERSION          PIC S9(9) BINARY VALUE 1.                                     2171
COPIED      **    Options that control the action of MQPUT and MQPUT1                                     2172
COPIED          15 MQPMO-OPTIONS          PIC S9(9) BINARY VALUE 0.                                     2173
COPIED      **    Reserved                                                                                2174
COPIED          15 MQPMO-TIMEOUT          PIC S9(9) BINARY VALUE -1.                                    2175
COPIED      **    Object handle of input queue                                                            2176
COPIED          15 MQPMO-CONTEXT          PIC S9(9) BINARY VALUE 0.                                     2177
COPIED      **    Reserved                                                                                2178
COPIED          15 MQPMO-KNOWNDESTCOUNT   PIC S9(9) BINARY VALUE 0.                                     2179
COPIED      **    Reserved                                                                                2180
COPIED          15 MQPMO-UNKNOWNDESTCOUNT PIC S9(9) BINARY VALUE 0.                                     2181
COPIED      **    Reserved                                                                                2182
COPIED          15 MQPMO-INVALIDDESTCOUNT PIC S9(9) BINARY VALUE 0.                                     2183
COPIED      **    Resolved name of destination queue                                                      2184
COPIED          15 MQPMO-RESOLVEDQNAME    PIC X(48) VALUE SPACES.                                       2185
COPIED      **    Resolved name of destination queue manager                                              2186
COPIED          15 MQPMO-RESOLVEDQMGRNAME PIC X(48) VALUE SPACES.                                       2187
COPIED                                                                                                    2188
COPIED      ****************************************************************                              2189
COPIED      **  End of CMQPMOV                                            **                              2190
COPIED      ****************************************************************                              2191
      MAIN:                                                                                               2192
      * ------------------------------------------------------------ *                                    2193
      *                                                              *                                    2194
      * This section receives the names of the queue manager and the *                                    2195
      * queue from the PARM statement in the JCL. It opens the queue, *                                   2196
      * reads all the messages, and prints them                      *                                    2197
      *                                                              *                                    2198
      * This section uses the MQGET call with the BROWSE option to   *                                    2199
      * ensure that the data is not removed from the queue           *                                    2200
      *                                                              *                                    2201
      * ------------------------------------------------------------ *                                    2202
      *                                                                                                   2203
      *    Open the print file, initialize the fields for the                                             2204

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    46
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
      *    header date and the page number, and print the first                                           2205
      *    line of the header                                                                             2206
      *                                                                                                   2207
      *                                                                                                   2208
           MOVE DYLDATE4 TO  W01_DATE                                                                     2209
      *                                                                                                   2210
           PERFORM PRINT_HDR1                                                                             2211
      *                                                                                                   2212
      *                                                                                                   2213
      *    Retrieved Queue Manager Name and Queue Name from JCL PARM statement                            2214
      *                                                                                                   2215
      *    Following fields are initialized                                                               2216
      *      PARMINFO =  Data retrieved from PARM statement                                               2217
      *      INX      =  Index used to find delimiter                                                     2218
      *      INZ      =  Index used to calculate size of PARM value                                       2219
      *      INY      =  Index used to determine number of PARMS retrieved                                2220
      *                                                                                                   2221
      *                                                                                                   2222
           MOVE DYLPARM    TO PARMINFO                                                                    2223
           MOVE 0          TO INX                                                                         2224
           MOVE 0          TO INZ                                                                         2225
           MOVE 1          TO INY                                                                         2226
      *                                                                                                   2227
      *    Separate into the relevant fields any data passed in the                                       2228
      *    PARM statement. The fields are separated by a comma. The last field                            2229
      *    ends with a blank.                                                                             2230
      *                                                                                                   2231
```

Figure 22 MQGET Sample Compiled Listing (Page 28 of 35)

```
    DOWHILE INX LE 60 AND INY LE 2                                              2232
      IF PARMDATA(INX) EQ ',' OR PARMDATA(INX) EQ ' '                           2233
        PARMLEN = INX - INZ             ;Length of PARM data value              2234
        CASE INY                        ;Determine current PARM field           2235
          WHEN EQ 1                     ;Queue Manager Name?                    2236
           MOVE PARMDATA(INZ)                                                   2237
            TO W02_MQM LENGTH PARMLEN   ;Save Queue Manager Name                2238
          WHEN EQ 2                     ;Queue Name?                            2239
           MOVE PARMDATA(INZ)                                                   2240
            TO W02_OBJECT LENGTH PARMLEN ;Save Queue Name                       2241
        ENDCASE                                                                 2242
        INY = INY + 1                   ;Next PARM data field number            2243
      ENDIF                                                                     2244
      IF PARMDATA(INX) NE ' '           ;Not the last PARM data field           2245
        IF PARMDATA(INX) EQ ','         ;End of current PARM data               2246
          INZ = INX + 1                 ;Start of new PARM data field           2247
        ENDIF                                                                   2248
        INX = INX + 1                   ;Check next parameter byte              2249
      ENDIF                                                                     2250
    ENDDO                                                                       2251
*                                                                               2252
*   If no data was passed, create a message, print it, and                      2253

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    47
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
*   exit                                                                        2254
*                                                                               2255
     IF INX GT 60                                                               2256
       MOVE W04_MESSAGE_1 TO W00_PRINT_DATA                                     2257
       PERFORM PRINT_LINE                                                       2258
       MOVE W06_CSQ4_WARNING TO DYLRETURN                                       2259
       GOTO MAIN_END                                                            2260
     ENDIF                                                                      2261
*                                                                               2262
*   Move the data (spaces if nothing is entered) into the                       2263
*   relevant API parameter fields                                               2264
*                                                                               2265
     MOVE W02_MQM    TO QMGR                                                    2266
*                                                                               2267
*   Move the data (spaces if nothing is entered) into the                       2268
*   relevant print fields                                                       2269
*                                                                               2270
     MOVE W02_MQM    TO W01_MQM_NAME                                            2271
     MOVE W02_OBJECT TO W01_QUEUE_NAME                                          2272
*                                                                               2273
*   Print a message if the queue manager name is missing, the                   2274
*   default queue manager will be used                                          2275
*                                                                               2276
    IF W02_MQM EQ SPACES OR W02_MQM EQ LOW_VALUES                               2277
      MOVE W04_MESSAGE_2 TO W00_PRINT_DATA                                      2278
      PERFORM PRINT_LINE                                                        2279
    ENDIF                                                                       2280
*                                                                               2281
*   Print a message if the queue name is missing and exit from                  2282
*   program                                                                     2283
*                                                                               2284
    IF W02_OBJECT EQ SPACES OR W02_OBJECT EQ LOW_VALUES                         2285
      MOVE W04_MESSAGE_3 TO W00_PRINT_DATA                                      2286
      PERFORM PRINT_LINE                                                        2287
      MOVE W06_CSQ4_WARNING TO DYLRETURN                                        2288
      GOTO MAIN_END                                                             2289
    ENDIF                                                                       2290
*                                                                               2291
*   Print the remaining header lines                                            2292
*                                                                               2293
    PERFORM PRINT_HDR2                                                          2294
*                                                                               2295
*   Connect to the specified queue manager.                                     2296
*                                                                               2297
    MQCONN   QMGR                   ;Queue Manager Name                         2298
             HCONN                  ;Connection Handle                          2299
             COMPCODE               ;Completion Code                            2300
             REASON                 ;Reason Code                                2301
*                                                                               2302

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    48
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
*   Test the output of the connect call.  If the call failed,                   2303
*   print an error message showing the completion code and                      2304
*   reason code                                                                 2305
*                                                                               2306
    IF COMPCODE NE MQCC_OK                                                      2307
      MOVE 'CONNECT'     TO W04_MSG4_TYPE                                       2308
      MOVE COMPCODE      TO COMP_CODE                                           2309
      MOVE REASON        TO REASON_CODE                                         2310
```

Figure 22 MQGET Sample Compiled Listing (Page 29 of 35)

```
       EDIT COMP_CODE   INTO W04_MSG4_COMPCODE USING Z                          2311
       EDIT REASON_CODE INTO W04_MSG4_REASON USING Z                           2312
       MOVE W04_MESSAGE_4 TO W00_PRINT_DATA                                    2313
       PERFORM PRINT_LINE                                                      2314
       MOVE W06_CSQ4_ERROR TO DYLRETURN                                        2315
       GOTO MAIN_END                                                           2316
    ENDIF                                                                      2317
*                                                                              2318
*   Initialize the object descriptor (MQOD) control block.                     2319
*   (The copy file initializes all the other fields)                           2320
*                                                                              2321
    MOVE MQOT_Q           TO MQOD_OBJECTTYPE                                   2322
    MOVE W02_OBJECT       TO MQOD_OBJECTNAME                                   2323
*                                                                              2324
*   Initialize the working storage fields required to open                     2325
*   the queue                                                                  2326
*                                                                              2327
*     HOPTIONS is set to open the queue for browsing                           2328
*     HOBJ     is set by the MQOPEN call and is used by the                    2329
*              MQGET and MQCLOSE calls                                         2330
*                                                                              2331
    MOVE MQOO_BROWSE      TO HOPTIONS                                          2332
*                                                                              2333
*   Open the queue.                                                            2334
*                                                                              2335
    MQOPEN   HCONN                ;Connection Handle                           2336
             MQOD                 ;Object Description                          2337
             HOPTIONS             ;Control Options                             2338
             HOBJ                 ;Object Handle                               2339
             COMPCODE             ;Completion Code                             2340
             REASON               ;Reason Code                                 2341
*                                                                              2342
*   Test the output of the open call.  If the call failed, print               2343
*   an error message showing the completion code and reason code               2344
*                                                                              2345
    IF COMPCODE NE MQCC_OK                                                     2346
       MOVE 'OPEN'        TO W04_MSG4_TYPE                                     2347
       MOVE COMPCODE      TO W04_MSG4_COMPCODE                                 2348
       MOVE REASON        TO W04_MSG4_REASON                                   2349
       MOVE W04_MESSAGE_4 TO W00_PRINT_DATA                                    2350
       PERFORM PRINT_LINE                                                      2351

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    49
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
       MOVE W06_CSQ4_ERROR TO DYLRETURN                                        2352
       GOTO DISCONNECT                                                         2353
    ENDIF                                                                      2354
*                                                                              2355
*   No need to change the Message Descriptor (MQMD) control                    2356
*   block because the copy file initializes all the fields                     2357
*                                                                              2358
*   Initialize the Get Message Options (MQGMO) control block.                  2359
*   (The copy file initializes all the other fields)                           2360
*                                                                              2361
    MQGMO_OPTIONS = MQGMO_NO_WAIT + MQGMO_ACCEPT_TRUNCATED_MSG                 2362
    MQGMO_OPTIONS = MQGMO_OPTIONS + MQGMO_BROWSE_FIRST                         2363
*                                                                              2364
*   Make the first get call outside the loop because this call                 2365
*   uses the BROWSE-FIRST option                                               2366
*                                                                              2367
    MQGET    HCONN                ;Connection Handle                           2368
             HOBJ                 ;Object Handle                               2369
             MQMD                 ;Messae Descriptor Attributes                2370
             MQGMO                ;Retrieval Options                           2371
             BUFFERLEN            ;Message Buffer Length                       2372
             BUFFER               ;Message Buffer                              2373
             DATALEN              ;Message Buffer's Data Length                2374
             COMPCODE             ;Completion Code                             2375
             REASON               ;Reason Code                                 2376
*                                                                              2377
*   Test the output of the get call using the PERFORM loop                     2378
*   that follows.                                                              2379
*                                                                              2380
*   Change the MQGMO Options field to BROWSE-NEXT.                             2381
*                                                                              2382
    MQGMO_OPTIONS = MQGMO_NO_WAIT + MQGMO_ACCEPT_TRUNCATED_MSG                 2383
    MQGMO_OPTIONS = MQGMO_OPTIONS + MQGMO_BROWSE_NEXT                          2384
*                                                                              2385
*   Loop until the get call fails                                              2386
*   - we test for call not successful and the one condition                    2387
*     after which we want to continue within the loop                          2388
*     (the received message has been truncated)                                2389
*                                                                              2390
    DOWHILE COMPCODE EQ MQCC_OK OR                                             2391
           (COMPCODE EQ MQCC_WARNING AND                                       2392
```

Figure 22 MQGET Sample Compiled Listing (Page 30 of 35)

```
            REASON EQ MQRC_TRUNCATED_MSG_ACCEPTED)                                  2393
*                                                                                   2394
*       Increment the relative message number.  Move the message                    2395
*       number and the first 80 bytes of the message data into                      2396
*       the print line                                                              2397
*                                                                                   2398
        W00_MESSAGE_COUNT = W00_MESSAGE_COUNT + 1                                   2399
        MOVE W00_MESSAGE_COUNT TO MESSAGE_COUNT                                     2400

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    50
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
        MOVE DATALEN           TO DATA_LENGTH                                       2401
        EDIT MESSAGE_COUNT   INTO W01_MESSAGE_NUMBER USING Z                        2402
        EDIT DATA_LENGTH     INTO W01_MESSAGE_LENGTH USING Z                        2403
        MOVE BUFFER            TO W01_DATA                                          2404
        MOVE W01_REPORT_LINE   TO W00_PRINT_DATA                                    2405
*                                                                                   2406
*       Print the message line                                                      2407
*                                                                                   2408
        PERFORM PRINT_LINE                                                          2409
*                                                                                   2410
*********************************************************************             2411
*    MQMD-MSGID and MQMD-CORRELID are input/output fields that    *                 2412
*    are filled and read by MQGET.  Clear them before the next    *                 2413
*    MQGET call to ensure that all messages are retrieved.        *                 2414
*********************************************************************             2415
*                                                                                   2416
        MOVE MQMI_NONE TO MQMD_MSGID                                                2417
        MOVE MQCI_NONE TO MQMD_CORRELID                                             2418
*                                                                                   2419
*    Clear the message data field before the next get call to                       2420
*    ensure that no old data remains if the next line is shorter                    2421
*                                                                                   2422
        MOVE SPACES TO BUFFER                                                       2423
*                                                                                   2424
*       Get the next message                                                        2425
*                                                                                   2426
        MQGET     HCONN              ;Connection Handle                             2427
                  HOBJ               ;Object Handle                                 2428
                  MQMD               ;Messae Descriptor Attributes                  2429
                  MQGMO              ;Retrieval Options                             2430
                  BUFFERLEN          ;Message Buffer Length                         2431
                  BUFFER             ;Message Buffer                                2432
                  DATALEN            ;Message Buffer's Data Length                  2433
                  COMPCODE           ;Completion Code                               2434
                  REASON             ;Reason Code                                   2435
                  ERRORMSG           ;Error Message                                 2436
*                                                                                   2437
*       Test the output of the MQGET call at the top of the loop.                   2438
*       Exit the loop if an error occurs                                            2439
*                                                                                   2440
           ENDDO                                                                    2441
*                                                                                   2442
*    Test the output of the get call                                                2443
*                                                                                   2444
*    When the loop reaches the end of the messages, the                             2445
*    completion code is MQCC-FAILED and the reason code                             2446
*    is MQRC-NO-MSG-AVAILABLE                                                       2447
*                                                                                   2448
*    If the call failed for any other reason,                                       2449

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    51
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
*    print an error message showing the completion code and                         2450
*    reason code                                                                    2451
*                                                                                   2452
     IF COMPCODE EQ MQCC_FAILED AND                                                 2453
        REASON EQ MQRC_NO_MSG_AVAILABLE                                             2454
*                                                                                   2455
        MOVE W04_MESSAGE_0 TO W00_PRINT_DATA                                        2456
*                                                                                   2457
     ELSE                                                                           2458
        MOVE 'GET'         TO W04_MSG4_TYPE                                         2459
        MOVE COMPCODE      TO W04_MSG4_COMPCODE                                     2460
        MOVE REASON        TO W04_MSG4_REASON                                       2461
        MOVE W04_MESSAGE_4 TO W00_PRINT_DATA                                        2462
     ENDIF                                                                          2463
*                                                                                   2464
     PERFORM PRINT_LINE                                                             2465
*                                                                                   2466
* Close the queue                                                                   2467
*                                                                                   2468
     MOVE MQCO_NONE TO HOPTIONS                                                     2469
*                                                                                   2470
     MQCLOSE      HCONN             ;Connection Handle                             2471
```

Figure 22 MQGET Sample Compiled Listing (Page 31 of 35)

```
                HOBJ                ;Object Handle                                  2472
                HOPTIONS            ;Control Options                                2473
                COMPCODE            ;Completion Code                                2474
                REASON              ;Reason Code                                    2475
*                                                                                   2476
*    Test the output of the MQCLOSE call.  If the call failed,                      2477
*    print an error message showing the completion code and reason                  2478
*    code                                                                           2479
*                                                                                   2480
     IF COMPCODE NE MQCC_OK                                                        2481
        MOVE 'CLOSE'       TO W04_MSG4_TYPE                                        2482
        MOVE COMPCODE      TO W04_MSG4_COMPCODE                                    2483
        MOVE REASON        TO W04_MSG4_REASON                                      2484
        MOVE W04_MESSAGE_4 TO W00_PRINT_DATA                                       2485
        PERFORM PRINT_LINE                                                         2486
        MOVE W06_CSQ4_ERROR TO DYLRETURN                                           2487
     ENDIF                                                                         2488
*                                                                                   2489
DISCONNECT:                                                                        2490
*                                                                                   2491
* Disconnect from the queue manager                                                 2492
*                                                                                   2493
     MQDISC    HCONN               ;Connection Handle                              2494
               COMPCODE            ;Completion Code                                2495
               REASON              ;Reason Code                                    2496
*                                                                                   2497
*    Test the output of the disconnect call.  If the call failed,                   2498

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    52
1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
*    print an error message showing the completion code and                         2499
*    reason code                                                                    2500
*                                                                                   2501
     IF COMPCODE NE MQCC_OK                                                        2502
        MOVE 'DISCONNECT'  TO W04_MSG4_TYPE                                        2503
        MOVE COMPCODE      TO W04_MSG4_COMPCODE                                    2504
        MOVE REASON        TO W04_MSG4_REASON                                      2505
        MOVE W04_MESSAGE_4 TO W00_PRINT_DATA                                       2506
        MOVE W06_CSQ4_ERROR TO DYLRETURN                                           2507
        PERFORM PRINT_LINE                                                         2508
     ENDIF                                                                         2509
*                                                                                   2510
MAIN_END:                                                                          2511
*                                                                                   2512
*    Stop the program                                                               2513
*                                                                                   2514
     STOP                                                                          2515
*                                                                                   2516
*********************************************************************               2517
*                                                                                   2518
PRINT_LINE:                                                                        2519
*                                                                                   2520
*    This section prints all data lines produced by the program                     2521
*                                                                                   2522
*    If the maximum number of lines for a page has been printed,                    2523
*    start a new page                                                               2524
*                                                                                   2525
     IF W00_LINE_COUNT GT W00_MAX_LINES                                            2526
        PERFORM PRINT_HDR1                                                         2527
        PERFORM PRINT_HDR2                                                         2528
     ENDIF                                                                         2529
*                                                                                   2530
     MOVE W00_PRINT_DATA TO PRINT_DATA                                             2531
     MOVE SPACE TO CARRIAGE_CONTROL                                                2532
     WRITE SYSOUT                                                                  2533
*                                                                                   2534
     W00_LINE_COUNT = W00_LINE_COUNT + 1                                           2535
*                                                                                   2536
*                                                                                   2537
*********************************************************************               2538
*                                                                                   2539
PRINT_HDR1:                                                                        2540
*                                                                                   2541
*    This section prints the first line of the report.                              2542
*    This is separate from the section that prints the other                        2543
*    header lines because the first line is needed every time                       2544
*    the program runs                                                               2545
*                                                                                   2546
     W00_PAGE_NUMBER = W00_PAGE_NUMBER + 1                                         2547

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    53
1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
     MOVE W00_PAGE_NUMBER TO W01_PAGE                                              2548
     MOVE W01_HEADER_1 TO PRINT_DATA                                               2549
     MOVE '1'  TO  CARRIAGE_CONTROL                                                2550
```

Figure 22 MQGET Sample Compiled Listing (Page 32 of 35)

```
    WRITE SYSOUT
*
    MOVE 1 TO W00_LINE_COUNT
*
*
*******************************************************************
*
PRINT_HDR2:
*
*   This section prints the remaining header lines
*
    MOVE W01_HEADER_2 TO PRINT_DATA
    MOVE '0' TO CARRIAGE_CONTROL
    WRITE SYSOUT
    W00_LINE_COUNT = W00_LINE_COUNT + 2
*
    MOVE W01_HEADER_3 TO PRINT_DATA
    MOVE SPACE TO CARRIAGE_CONTROL
    WRITE SYSOUT
    W00_LINE_COUNT = W00_LINE_COUNT + 1
*
    MOVE W01_HEADER_4 TO PRINT_DATA
    MOVE '0' TO CARRIAGE_CONTROL
    WRITE SYSOUT
    W00_LINE_COUNT = W00_LINE_COUNT + 2
*
    MOVE W01_HEADER_5 TO PRINT_DATA
    MOVE SPACE TO CARRIAGE_CONTROL
    WRITE SYSOUT
    W00_LINE_COUNT = W00_LINE_COUNT + 1
*
    MOVE W01_HEADER_6 TO PRINT_DATA
    MOVE SPACE TO CARRIAGE_CONTROL
    WRITE SYSOUT
    W00_LINE_COUNT = W00_LINE_COUNT + 1
*
    MOVE SPACES      TO PRINT_DATA
    MOVE SPACE TO CARRIAGE_CONTROL
    WRITE SYSOUT
    W00_LINE_COUNT = W00_LINE_COUNT + 1

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    54
              C R O S S   R E F E R E N C E
                                    A       T D
                                    R       Y E
          DATANAME                  E       P C
           OR TAG               LOCN A  SIZE  E . DEFN              REFERENCES

BUFFER                          2905 P 32767 CH    309 -  2368 2404 2423 2427
BUFFERLEN                       2897 P     4 BI 0  307 -  2368 2427
CARRIAGE_CONTROL                   1 J     1 CH    147 -  2532 2550 2563 2568 2573 2578
                                                          2583 2588
COMP_CODE                        603 P     4 PD 0  154 -  2309 2311
COMPCODE                        2681 P     4 BI 0  301 -  2298 2307 2309 2336 2346 2348
                                                          2368 2391 2391 2427 2453 2460
                                                          2471 2481 2483 2494 2502 2504
DATA_LENGTH                      599 P     4 PD 0  153 -  2401 2403
DATALEN                         2913 Q     4 BI 0  314 -  2368 2401 2427
DISCONNECT:                                       2490 -  2353
DYLDATE4                         497 P    10 CH    DYL -  2209
DYLPARM                           41 U    60 CH    DYL -  2223
DYLRETURN                       RTC  U     2 BI    DYL -  2259 2288 2315 2352 2487 2507
ERRORMSG                        2697 P   200 CH    305 -  2427
HCONN                           2657 P     4 BI 0  295 -  2298 2336 2368 2427 2471 2494
HOBJ                            2665 P     4 BI 0  297 -  2336 2368 2427 2471
HOPTIONS                        2673 P     4 BI 0  299 -  2332 2336 2469 2471
INX                             INX  U     2 BI    DYL -  2224 2232 2234 2247 2249 2249
                                                          2256
INY                             INY  U     2 BI    DYL -  2226 2232 2235 2243 2243
INZ                             INZ  U     2 BI    DYL -  2225 2234 2247
LOW_VALUES                       547 P    48 CH    151 -  2277 2285
MAIN_END:                                         2511 -  2260 2289 2316
MESSAGE_COUNT                    595 P     4 PD 0  152 -  2400 2402
MQCC_FAILED                     5958 R     4 BI 0 1299 -  2453
MQCC_OK                         5950 R     4 BI 0 1297 -  2307 2346 2391 2481 2502
MQCC_WARNING                    5954 R     4 BI 0 1298 -  2391
MQCI_NONE                       4271 R    24 CH    745 -  2418
MQCO_NONE                       4942 R     4 BI 0 1014 -  2469
MQGMO                           8385 R   100 CH   2033 -  2368 2427
MQGMO_ACCEPT_TRUNCATED_MSG      3504 R     4 BI 0  484 -  2362 2383
MQGMO_BROWSE_FIRST              3480 R     4 BI 0  478 -  2363
MQGMO_BROWSE_NEXT               3484 R     4 BI 0  479 -  2384
MQGMO_NO_WAIT                   3452 R     4 BI 0  471 -  2362 2383
MQGMO_OPTIONS                   8393 R     4 BI 0 2039 -  2362 2363 2363 2383 2384 2384
MQMD                            8017 R   364 CH   1940 -  2368 2427
```

Figure 22   MQGET Sample Compiled Listing (Page 33 of 35)

```
MQMD_CORRELID                  8089 R    24 CH   1966 -  2418
MQMD_MSGID                     8065 R    24 CH   1964 -  2417
MQMI_NONE                      4247 R    24 CH    742 -  2417
MQOD                           8489 R   336 CH   2097 -  2336
MQOD_OBJECTNAME                8501 R    48 CH   2105 -  2323
MQOD_OBJECTTYPE                8497 R     4 BI 0 2103 -  2322

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    55
            C R O S S   R E F E R E N C E
                                  A       T D
                                  R       Y E
           DATANAME               E       P C
            OR TAG             LOCN A  SIZE  E . DEFN            REFERENCES

MQOO_BROWSE                    5506 R     4 BI 0 1175 -  2332
MQOT_Q                         4591 R     4 BI 0  849 -  2322
MQRC_NO_MSG_AVAILABLE          6094 R     4 BI 0 1335 -  2453
MQRC_TRUNCATED_MSG_ACCEPTED    6258 R     4 BI 0 1376 -  2391
PARMDATA                        617 P     1 CH    158 -  2233 2233 2237 2240 2245 2246
PARMINFO                        617 P    60 CH    157 -  2223
PARMLEN                         545 P     2 BI 0  150 -  2234 2237 2240
PRINT_DATA                        2 J   132 CH    148 -  2531 2549 2562 2567 2572 2577
                                                        2582 2587
PRINT_HDR1:                                     2540 -  2211 2527
PRINT_HDR2:                                     2558 -  2294 2528
PRINT_LINE:                                     2519 -  2258 2279 2287 2314 2351 2409
                                                        2465 2486 2508
QMGR                           2609 P    48 CH    293 -  2266 2298
REASON                         2689 P     4 BI 0  303 -  2298 2310 2336 2349 2368 2391
                                                        2427 2453 2461 2471 2484 2494
                                                        2505
REASON_CODE                     607 P     4 PD 0  155 -  2310 2312
SPACE                           481 P     1 CH    DYL -  2532 2568 2578 2583 2588
SPACES                          481 P     1 CH    DYL -  2277 2285 2423 2587
W00_LINE_COUNT                  689 P     2 BI 0  166 -  2526 2535 2535 2553 2565 2565
                                                        2570 2570 2575 2575 2580 2580
                                                        2585 2585 2590 2590
W00_MAX_LINES                   681 P     2 BI 0  165 -  2526
W00_MESSAGE_COUNT               705 P     4 BI 0  168 -  2399 2399 2400
W00_PAGE_NUMBER                 697 P     2 BI 0  167 -  2547 2547 2548
W00_PRINT_DATA                  721 P   132 CH    173 -  2257 2278 2286 2313 2350 2405
                                                        2456 2462 2485 2506 2531
W01_DATA                       1709 P    80 CH    219 -  2404
W01_DATE                        867 P    10 CH    179 -  2209
W01_HEADER_1                    857 P   132 CH    177 -  2549
W01_HEADER_2                    993 P   132 CH    187 -  2562
W01_HEADER_3                   1129 P   132 CH    193 -  2567
W01_HEADER_4                   1265 P   132 CH    199 -  2572
W01_HEADER_5                   1401 P   132 CH    202 -  2577
W01_HEADER_6                   1537 P   132 CH    206 -  2582
W01_MESSAGE_LENGTH             1699 P     9 CH    217 -  2403
W01_MESSAGE_NUMBER             1689 P     9 CH    215 -  2402
W01_MQM_NAME                   1047 P    48 CH    191 -  2271
W01_PAGE                        975 P     4 CH    185 -  2548
W01_QUEUE_NAME                 1183 P    48 CH    197 -  2272
W01_REPORT_LINE                1673 P   132 CH    213 -  2405
W02_MQM                        1809 P    48 CH    224 -  2237 2266 2271 2277 2277

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    56
            C R O S S   R E F E R E N C E
                                  A       T D
                                  R       Y E
           DATANAME               E       P C
            OR TAG             LOCN A  SIZE  E . DEFN            REFERENCES

W02_OBJECT                     1857 P    48 CH    225 -  2240 2272 2285 2285 2323
W04_MESSAGE_0                  1905 P   132 CH    229 -  2456
W04_MESSAGE_1                  2041 P   132 CH    234 -  2257
W04_MESSAGE_2                  2177 P   132 CH    239 -  2278
W04_MESSAGE_3                  2313 P   132 CH    244 -  2286
W04_MESSAGE_4                  2449 P   132 CH    248 -  2313 2350 2462 2485 2506
W04_MSG4_COMPCODE              2524 P     9 CH    255 -  2311 2348 2460 2483 2504
W04_MSG4_REASON                2548 P     9 CH    257 -  2312 2349 2461 2484 2505
W04_MSG4_TYPE                  2494 P    10 CH    252 -  2308 2347 2459 2482 2503
W06_CSQ4_ERROR                 2601 P     4 NU 0  264 -  2315 2352 2487 2507
W06_CSQ4_WARNING               2593 P     4 NU 0  263 -  2259 2288

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    57
1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
DEFAULT OPTIONS SPECIFIED ARE:

   ENVIRONMENT IS MVS
   CPUID  2084       C0851

   ASALINE(N/), BATCHIQ(IIBATCH/), CBXSIGN(N/), CENTNEW(75/), CENTRY1(/), CENTRY2(/), COBEDIT(A/),
```

Figure 22 MQGET Sample Compiled Listing (Page 34 of 35)

```
   COBENV(N/), COMPERR(N/), COB2NR(N/), COMPWRK(1500K/), CURNCY($/), DATAIRN(N/), DECIML9(E/),
   DELIM(/), DLMFRST(Y/), DUPCBNM(N/), DYLVARP(N/), DYL4YR(N/), EDP1ZERO(N/), EDSUPR(N/),
   EURODAT(N/), EURONUM(N/), EXCEL(N/), EXCLPAT(N/), EXPRERR(N/), FREEMEM(1000K/),
   FREEZDD(SYS280FZ/), GETMAX(2500K/), KWDLT(N/), LE(Y/), LIBRBUF(60K/), LPFUNLMT(N/), LSTSIMX(N/),
   LITRFROM(N/), LITRZERO(N/), MACHORG(N/), MAXDNLN(50/), MAXDYLF(175/), MNAMENU(N/), NAMEHDR(/),
   NDVRCOM(N/), NDVRENV(/), NODLETE(N/), NOVSOIO(N/), NOPOWRT(N/), NOSRTAB(N/), NOTOTAL(N/),
   NOVSIO(N/), NUMCHAR(N/), NUMPD(N/), OPTPRDG(Y/), OPTPRER(N/), OUTFILE(N/), PANVBUF(60K/),
   PDSREPL(N/), PGLINER(55/), PGLINES(55/), PROGMOD(XREFREF,CONVENTIONAL,EXP/,STRUCT2), PRTCIRS(N/),
   PRTZERO(N/), QLF(N/), RANDMPCT(N/), RDYONLY(N/), RESRWRD(N/), RETCODE(Y/), RPTASA(N/),
   RPTDDNM(SYS280R/), RPTXPAG(N/), SORTDEV(SYSDA/), SORTDYN(N/), SORTMEM(36K,100K,700K,100K/),
   SORTNAM(SORT/), SSMASK(N/), STRUCGO(N/), SUBRADD(Y/), SUPCOBW(Y/), SUPRESQ(N/), SUP182W(N/),
   SYSBLOK(N/), TIMESEP(D/), TABLEHI(N/), VSAMCAT(Y/), VDUPARND(N/), VSAMMSG(N/), XREF$(N/),
   ZDIVAB(O/)

COMPUTER ASSOCIATES       VISION:RESULTS  6.0 *              DATE  09/20/05  18.21.32   PAGE     1
         1         2         3         4         5         6         7         8         9
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
VISION:RESULTS CONTROL TOTALS

FILE       RECORD         CHARACTER     BLOCK     DROPPED     REWRITTEN      INSERTED      ERASED
 ID         COUNT           COUNT       COUNT   BLOCK COUNT  RECORD COUNT  RECORD COUNT RECORD COUNT

SYSOUT             9             1,197


                            RECORDS                  PAGES

FILE PRINT                                              1

REPORT PRINT

FIXED BLANK COUNT

FIXED DECIMAL DIVIDE


    RETURN CODE-0000


   ******************************************************
   *                                                    *
   *  (C) 2005 COMPUTER ASSOCIATES INTERNATIONAL, INC.  *
   *                                                    *
   ******************************************************

09/20/2005                                SAMPLE QUEUE REPORT                                PAGE 0001
                        QUEUE MANAGER NAME : CSQ1
                                QUEUE NAME : VISION.RESULTS.TEST.QUEUE
        RELATIVE
         MESSAGE   MESSAGE
         NUMBER    LENGTH ---------------------------------- MESSAGE DATA ---------------------------------
             1         40  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                                ********** END OF REPORT **********
```

Figure 22 MQGET Sample Compiled Listing (Page 35 of 35)

MQPUT Sample Compiled Listing

As shown in the compiled listing (Figure 23), the MQPUT program is executed using the following JCL PARM parameter as input:

```
PARM='UA=CSQ1,VISION.RESULTS.TEST.QUEUE,1,A,40,N'
```

```
COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE     1
         1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
         OPTION STRUCTURED2                                                                        1
         * ************************************************************ *                        2
         *                                                              *                        3
         *  Description      : Sample program to put a number of        *                        4
         *                     messages to a queue.                     *                        5
         *                                                              *                        6
         *  Function         : This program writes a specified number of *                        7
         *                     messages to a specified queue in a       *                        8
```

Figure 23 MQPUT Sample Compiled Listing (Page 1 of 32)

```
*                  specified queue manager. The messages    *                                  9
*                  would consist of a one byte character    *                                 10
*                  that would be permeated through the      *                                 11
*                  entire message.                          *                                 12
*                                                           *                                 13
*                  The queue manager, the queue name, the   *                                 14
*                  number of messages, the length of the    *                                 15
*                  messages, the persistence of the         *                                 16
*                  messages, and the one byte pad character *                                 17
*                  that would be used for the message would *                                 18
*                  be passed as input via the JCL PARM      *                                 19
*                  parameter.                               *                                 20
*                                                           *                                 21
*                  Note: Since, this is a test program      *                                 22
*                        the persistence of the messages    *                                 23
*                        should be set to N.                *                                 24
*                                                           *                                 25
* Limitation      : Maximum message length set at 32767.    *                                 26
*                                                           *                                 27
* ********************************************************* *                                 28
*                                                           *                                 29
*                    Program  Logic                         *                                 30
*                    --------------                         *                                 31
*                                                           *                                 32
*                                                           *                                 33
* main                                                      *                                 34
* ----                                                      *                                 35
*                                                           *                                 36
*      Obtain the input data from                           *                                 37
*       PARM='UA=aaa,bbb,ccc,d,eee,f':                      *                                 38
*         - aaa is the name of the queue manager            *                                 39
*         - bbb is the name of the queue                    *                                 40
*         - ccc is the number of messages                   *                                 41
*         - d   is the message pad character                *                                 42
*         - eee is the length of message(s)                 *                                 43
*         - f   is the persistence of message(s)            *                                 44
*                                                           *                                 45
*   Move parameters into corresponding variables.           *                                 46
*   If parameters are invalid then                          *                                 47
*           Call USAGE_ERR and exit.                        *                                 48
*                                                           *                                 49

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE     2
1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
*   Display the parameters passed to the program.           *                                 50
*                                                           *                                 51
*   Connect to the queue manager.                           *                                 52
*   If connection failed then                               *                                 53
*           Call PRINT_MSG and exit                         *                                 54
*                                                           *                                 55
*   Open the specified message queue (MQOPEN).              *                                 56
*   If open failed then                                     *                                 57
*           Disconnect from queue manager                   *                                 58
*           Call PRINT_MSG and exit                         *                                 59
*   Endif.                                                  *                                 60
*                                                           *                                 61
*   Set the put message options.                            *                                 62
*   Loop while the messages are put to queue                *                                 63
*           Put message to queue (MQPUT)                    *                                 64
*           If put failed                                   *                                 65
*                   Call PRINT_MSG                          *                                 66
*                   Break from loop                         *                                 67
*           Endif                                           *                                 68
*   Endloop.                                                *                                 69
*   Display number of messages put to the queue.            *                                 70
*                                                           *                                 71
*   Close the message queue.                                *                                 72
*   If close failed then                                    *                                 73
*           Call PRINT_MSG.                                 *                                 74
*                                                           *                                 75
*   Disconnect from the queue manager.                      *                                 76
*   If disconnect failed then                               *                                 77
*           Call PRINT_MSG.                                 *                                 78
*                                                           *                                 79
*   Exit program.                                           *                                 80
*                                                           *                                 81
*                                                           *                                 82
*-----------------------------------------------------------*                                 83
*                                                           *                                 84
* USAGE_ERR                                                 *                                 85
* -----------                                               *                                 86
*                                                           *                                 87
*   Print message showing correct usage for program.        *                                 88
*                                                           *                                 89
*                                                           *                                 90
```

Figure 23 MQPUT Sample Compiled Listing (Page 2 of 32)

```
*-------------------------------------------------------------*
*                                                             *
*  PRINT_MSG                                                  *
*  --------------------                                       *
*                                                             *
*   Create error message and display.                         *
*                                                             *
*                                                             *

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05             PAGE     3
1-------------------VISION:RESULTS FREE FORM TEXT-------------------72          --------
* ************************************************************ *
* ------------------------------------------------------------ *
FILE XFILE DUMMY
WORKAREA
  PARMLEN           2  BI VALUE  0
  LOW_VALUES       48  CH VALUE LOWVALUES
  COMP_CODE         4  CH
  REASON_CODE       4  CH
  NUMBER_PUTS       4  CH
WORKAREA
  PARMINFO         60     VALUE NULL  ;JCL PARM parameter
  PARMDATA          1  1              ;PARM data one byte at a time
WORKAREA
       $COBOL
      * ------------------------------------------------------------ *
      *
      *    W00 - General work fields
      *
       01  W00-RETURN-CODE            PIC S9(4) BINARY VALUE ZERO.
       01  W00-NUMPUTS                PIC S9(9) BINARY VALUE 0.
       01  W00-ERROR-MESSAGE          PIC X(48) VALUE SPACES.
      *
      *    Data fields derived from the PARM field
      *
       01  W00-QMGR                   PIC X(48).
       01  W00-QNAME                  PIC X(48).
       01  W00-PADCHAR                PIC X(1) VALUE '*'.
       01  W00-MSGBUFFER.
         02  W00-MSGBUFFER-DATA       PIC X(32767).
         02  W00-MSGBUFFER-ARRAY REDEFINES W00-MSGBUFFER-DATA
                                      PIC X(1) OCCURS 32767 TIMES.
       01  W00-NUMMSGS-NUM            PIC 9(5) VALUE  0.
       01  W00-NUMMSGS-NUM-CHAR REDEFINES W00-NUMMSGS-NUM
                                      PIC X(5).
       01  W00-NUMMSGS                PIC S9(9) BINARY VALUE 1.
       01  W00-MSGLENGTH-NUM          PIC 9(5) VALUE 0.
       01  W00-MSGLENGTH-NUM-CHAR REDEFINES W00-MSGLENGTH-NUM
                                      PIC X(5).
       01  W00-PERSISTENCE            PIC X(1) VALUE 'N'.
           88 PERSISTENT     VALUE 'P'.
WARNING   DYL-182W    66/88/MISSING LEVEL - SKIPPING TO NEXT PERIOD
           88 NOT-PERSISTENT VALUE 'N'.
WARNING   DYL-182W    66/88/MISSING LEVEL - SKIPPING TO NEXT PERIOD
       $ECOBOL
*
*    The following copy files define API control blocks.
*
WORKAREA
COPY CMQWORK COBOL

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05             PAGE     4
1-------------------VISION:RESULTS FREE FORM TEXT-------------------72          --------
    COPIED      *****************************************************************
    COPIED      **                                                             **
    COPIED      **                WebSphere MQ for z/OS                        **
    COPIED      **                                                             **
    COPIED      **  FILE NAME:      CMQWORK                                    **
    COPIED      **                                                             **
    COPIED      **  DESCRIPTION:    Work Area Structures And Parms             **
    COPIED      **                                                             **
    COPIED      *****************************************************************
    COPIED      **                                                             **
    COPIED      **  FUNCTION:       This file declares work area structures and **
    COPIED      **                  parms, which are not found in other files.  **
    COPIED      **                                                             **
    COPIED      **                  This file also contains a copy of the COBOL **
    COPIED      **                  data file called CMQV which has an          **
    COPIED      **                  assortment of constants used by the MQIs.   **
    COPIED      **                  and other COBOL data structures.            **
    COPIED      **                                                             **
    COPIED      **  PROCESSOR:      COBOL                                      **
    COPIED      **                                                             **
    COPIED      *****************************************************************
    COPIED      * QUEUE MANAGER NAME
```

Figure 23   MQPUT Sample Compiled Listing (Page 3 of 32)

```
COPIED       01  QMGR             PIC X(48).                                  168
COPIED      * CONNECTION HANDLE                                                169
COPIED       01  HCONN            PIC S9(9) BINARY.                           170
COPIED      * OBJECT HANDLE                                                    171
COPIED       01  HOBJ             PIC S9(9) BINARY.                           172
COPIED      * OPTIONS THAT CONTROL THE ACTION                                  173
COPIED       01  HOPTIONS         PIC S9(9) BINARY.                           174
COPIED      * COMPLETION CODE                                                  175
COPIED       01  COMPCODE         PIC S9(9) BINARY.                           176
COPIED      * REASON CODE                                                      177
COPIED       01  REASON           PIC S9(9) BINARY.                           178
COPIED      * TEXT ERROR MESSAGE                                               179
COPIED       01  ERRORMSG         PIC X(200).                                 180
COPIED      * TOTAL LENGTH OF MESSAGE BUFFER                                   181
COPIED       01  BUFFERLEN        PIC S9(9) BINARY.                           182
COPIED      * MESSAGE BUFFER                                                   183
COPIED       01  BUFFER.                                                       184
COPIED          02  BUFFERDATA    PIC X(32767).                               185
COPIED          02  BUFFERARRAY REDEFINES BUFFERDATA                          186
COPIED                          PIC X(1) OCCURS 32767 TIMES.                  187
COPIED      * LENGTH OF THE DATA IN THE MESSAGE BUFFER                         188
COPIED       01  DATALEN          PIC S9(9) BINARY.                           189
COPIED      * COUNT OF ATTRIBUTE SELECTORS                                     190
COPIED       01  SELECTORCOUNT    PIC S9(9) BINARY.                           191
COPIED      * ARRAY OF ATTRIBUTE SELECTORS                                     192
COPIED       01  SELECTORSTABLE.                                              193
COPIED          02  SELECTORS     PIC S9(9) BINARY OCCURS 10 TIMES.           194

     COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    5
     1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED      * COUNT OF INTEGER ATTRIBUTES                                      195
COPIED       01  INTATTRCOUNT PIC S9(9) BINARY.                               196
COPIED      * ARRAY OF INTEGER ATTRIBUTES                                      197
COPIED       01  INTATTRSTABLE.                                               198
COPIED          02  INTATTRS  PIC S9(9) BINARY OCCURS 10 TIMES.               199
COPIED      * CHARACTER ATTRIBUTE BUFFER LENGTH                                200
COPIED       01  CHARATTRLENGTH PIC S9(9) BINARY.                             201
COPIED      * CHARACTER ATTRIBUTES                                             202
COPIED       01  CHARATTRS.                                                   203
COPIED          02  CHARATTRSDATA  PIC X(32767).                             204
COPIED          02  CHARATTRSARRAY REDEFINES CHARATTRSDATA                   205
COPIED                          PIC X(1) OCCURS 32767 TIMES.                  206
COPIED      ****************************************************************  207
COPIED      **  Values Related to MQCIH Structure                         **  208
COPIED      ****************************************************************  209
COPIED                                                                        210
COPIED      **   Structure Identifier                                         211
COPIED        10 MQCIH-STRUC-ID PIC X(4) VALUE 'CIH '.                        212
COPIED                                                                        213
COPIED      **   Structure Version Number                                     214
COPIED        10 MQCIH-VERSION-1       PIC S9(9) BINARY VALUE 1.             215
COPIED        10 MQCIH-VERSION-2       PIC S9(9) BINARY VALUE 2.             216
COPIED        10 MQCIH-CURRENT-VERSION PIC S9(9) BINARY VALUE 2.             217
COPIED                                                                        218
COPIED      **   Structure Length                                             219
COPIED        10 MQCIH-LENGTH-1       PIC S9(9) BINARY VALUE 164.           220
COPIED        10 MQCIH-LENGTH-2       PIC S9(9) BINARY VALUE 180.           221
COPIED        10 MQCIH-CURRENT-LENGTH PIC S9(9) BINARY VALUE 180.           222
COPIED                                                                        223
COPIED      **   Flags                                                        224
COPIED        10 MQCIH-NONE PIC S9(9) BINARY VALUE 0.                         225
COPIED                                                                        226
COPIED      **   Return Code                                                  227
COPIED        10 MQCRC-OK                   PIC S9(9) BINARY VALUE 0.       228
COPIED        10 MQCRC-CICS-EXEC-ERROR      PIC S9(9) BINARY VALUE 1.       229
COPIED        10 MQCRC-MQ-API-ERROR         PIC S9(9) BINARY VALUE 2.       230
COPIED        10 MQCRC-BRIDGE-ERROR         PIC S9(9) BINARY VALUE 3.       231
COPIED        10 MQCRC-BRIDGE-ABEND         PIC S9(9) BINARY VALUE 4.       232
COPIED        10 MQCRC-APPLICATION-ABEND    PIC S9(9) BINARY VALUE 5.       233
COPIED        10 MQCRC-SECURITY-ERROR       PIC S9(9) BINARY VALUE 6.       234
COPIED        10 MQCRC-PROGRAM-NOT-AVAILABLE PIC S9(9) BINARY VALUE 7.      235
COPIED        10 MQCRC-BRIDGE-TIMEOUT       PIC S9(9) BINARY VALUE 8.       236
COPIED        10 MQCRC-TRANSID-NOT-AVAILABLE PIC S9(9) BINARY VALUE 9.      237
COPIED                                                                        238
COPIED      **   Unit of Work Control                                         239
COPIED        10 MQCUOWC-ONLY     PIC S9(9) BINARY VALUE 273.               240
COPIED        10 MQCUOWC-CONTINUE PIC S9(9) BINARY VALUE 65536.             241
COPIED        10 MQCUOWC-FIRST    PIC S9(9) BINARY VALUE 17.                242
COPIED        10 MQCUOWC-MIDDLE   PIC S9(9) BINARY VALUE 16.                243

     COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    6
     1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQCUOWC-LAST     PIC S9(9) BINARY VALUE 272.               244
COPIED        10 MQCUOWC-COMMIT   PIC S9(9) BINARY VALUE 256.                245
COPIED        10 MQCUOWC-BACKOUT  PIC S9(9) BINARY VALUE 4352.              246
```

Figure 23 MQPUT Sample Compiled Listing (Page 4 of 32)

```
COPIED                                                                          247
COPIED      **   Get Wait Interval                                              248
COPIED        10 MQOGWI-DEFAULT PIC S9(9) BINARY VALUE -2.                      249
COPIED                                                                          250
COPIED      **   Link Type                                                      251
COPIED        10 MQCLT-PROGRAM     PIC S9(9) BINARY VALUE 1.                    252
COPIED        10 MQCLT-TRANSACTION PIC S9(9) BINARY VALUE 2.                    253
COPIED                                                                          254
COPIED      **   Output Data Length                                             255
COPIED        10 MQCODL-AS-INPUT PIC S9(9) BINARY VALUE -1.                     256
COPIED                                                                          257
COPIED      **   ADS Descriptor                                                 258
COPIED        10 MQCADSD-NONE      PIC S9(9) BINARY VALUE 0.                    259
COPIED        10 MQCADSD-SEND      PIC S9(9) BINARY VALUE 1.                    260
COPIED        10 MQCADSD-RECV      PIC S9(9) BINARY VALUE 16.                   261
COPIED        10 MQCADSD-MSGFORMAT PIC S9(9) BINARY VALUE 256.                  262
COPIED                                                                          263
COPIED      **   Conversational Task                                            264
COPIED        10 MQCCT-YES PIC S9(9) BINARY VALUE 1.                            265
COPIED        10 MQCCT-NO  PIC S9(9) BINARY VALUE 0.                            266
COPIED                                                                          267
COPIED      **   Task End Status                                                268
COPIED        10 MQCTES-NOSYNC  PIC S9(9) BINARY VALUE 0.                       269
COPIED        10 MQCTES-COMMIT  PIC S9(9) BINARY VALUE 256.                     270
COPIED        10 MQCTES-BACKOUT PIC S9(9) BINARY VALUE 4352.                    271
COPIED        10 MQCTES-ENDTASK PIC S9(9) BINARY VALUE 65536.                   272
COPIED                                                                          273
COPIED      **   Facility                                                       274
COPIED        10 MQCFAC-NONE PIC X(8) VALUE LOW-VALUES.                         275
COPIED                                                                          276
COPIED      **   Function                                                       277
COPIED        10 MQCFUNC-MQCONN PIC X(4) VALUE 'CONN'.                          278
COPIED        10 MQCFUNC-MQGET  PIC X(4) VALUE 'GET '.                          279
COPIED        10 MQCFUNC-MQINQ  PIC X(4) VALUE 'INQ '.                          280
COPIED        10 MQCFUNC-MQOPEN PIC X(4) VALUE 'OPEN'.                          281
COPIED        10 MQCFUNC-MQPUT  PIC X(4) VALUE 'PUT '.                          282
COPIED        10 MQCFUNC-MQPUT1 PIC X(4) VALUE 'PUT1'.                          283
COPIED        10 MQCFUNC-NONE   PIC X(4) VALUE SPACES.                          284
COPIED                                                                          285
COPIED      **   Start Code                                                     286
COPIED        10 MQCSC-START     PIC X(4) VALUE 'S   '.                         287
COPIED        10 MQCSC-STARTDATA PIC X(4) VALUE 'SD  '.                         288
COPIED        10 MQCSC-TERMINPUT PIC X(4) VALUE 'TD  '.                         289
COPIED        10 MQCSC-NONE      PIC X(4) VALUE SPACES.                         290
COPIED                                                                          291
COPIED                                                                          292
       COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE     7
       1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED      ******************************************************************  293
COPIED      **  Values Related to MQCNO Structure                           **  294
COPIED      ******************************************************************  295
COPIED                                                                          296
COPIED      **   Structure Identifier                                           297
COPIED        10 MQCNO-STRUC-ID PIC X(4) VALUE 'CNO '.                          298
COPIED                                                                          299
COPIED      **   Structure Version Number                                       300
COPIED        10 MQCNO-VERSION-1       PIC S9(9) BINARY VALUE 1.              301
COPIED        10 MQCNO-VERSION-2       PIC S9(9) BINARY VALUE 2.              302
COPIED        10 MQCNO-VERSION-3       PIC S9(9) BINARY VALUE 3.              303
COPIED        10 MQCNO-CURRENT-VERSION PIC S9(9) BINARY VALUE 3.              304
COPIED                                                                          305
COPIED      **   Connect Options                                                306
COPIED        10 MQCNO-STANDARD-BINDING        PIC S9(9) BINARY VALUE 0.      307
COPIED        10 MQCNO-FASTPATH-BINDING        PIC S9(9) BINARY VALUE 1.      308
COPIED        10 MQCNO-SERIALIZE-CONN-TAG-Q-MGR PIC S9(9) BINARY VALUE 2.     309
COPIED        10 MQCNO-SERIALIZE-CONN-TAG-QSG   PIC S9(9) BINARY VALUE 4.     310
COPIED        10 MQCNO-RESTRICT-CONN-TAG-Q-MGR  PIC S9(9) BINARY VALUE 8.     311
COPIED        10 MQCNO-RESTRICT-CONN-TAG-QSG    PIC S9(9) BINARY VALUE 16.    312
COPIED        10 MQCNO-NONE                     PIC S9(9) BINARY VALUE 0.     313
COPIED                                                                          314
COPIED      **   Queue-Manager Connection Tag                                   315
COPIED        10 MQCT-NONE PIC X(128) VALUE LOW-VALUES.                         316
COPIED                                                                          317
COPIED                                                                          318
COPIED      ******************************************************************  319
COPIED      **  Values Related to MQDLH Structure                           **  320
COPIED      ******************************************************************  321
COPIED                                                                          322
COPIED      **   Structure Identifier                                           323
COPIED        10 MQDLH-STRUC-ID PIC X(4) VALUE 'DLH '.                          324
COPIED                                                                          325
COPIED      **   Structure Version Number                                       326
COPIED        10 MQDLH-VERSION-1       PIC S9(9) BINARY VALUE 1.              327
COPIED        10 MQDLH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.              328
COPIED                                                                          329
```

Figure 23 MQPUT Sample Compiled Listing (Page 5 of 32)

```
COPIED                                                                          330
COPIED      ****************************************************************    331
COPIED      **  Values Related to MQGMO Structure                         **    332
COPIED      ****************************************************************    333
COPIED                                                                          334
COPIED      **   Structure Identifier                                           335
COPIED        10 MQGMO-STRUC-ID PIC X(4) VALUE 'GMO '.                          336
COPIED                                                                          337
COPIED      **   Structure Version Number                                       338
COPIED        10 MQGMO-VERSION-1       PIC S9(9) BINARY VALUE 1.              339
COPIED        10 MQGMO-VERSION-2       PIC S9(9) BINARY VALUE 2.              340
COPIED        10 MQGMO-VERSION-3       PIC S9(9) BINARY VALUE 3.              341

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE     8
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQGMO-CURRENT-VERSION PIC S9(9) BINARY VALUE 3.                342
COPIED                                                                          343
COPIED      **   Get-Message Options                                            344
COPIED        10 MQGMO-WAIT                    PIC S9(9) BINARY VALUE 1.     345
COPIED        10 MQGMO-NO-WAIT                 PIC S9(9) BINARY VALUE 0.     346
COPIED        10 MQGMO-SET-SIGNAL              PIC S9(9) BINARY VALUE 8.     347
COPIED        10 MQGMO-FAIL-IF-QUIESCING       PIC S9(9) BINARY VALUE 8192.  348
COPIED        10 MQGMO-SYNCPOINT               PIC S9(9) BINARY VALUE 2.     349
COPIED        10 MQGMO-SYNCPOINT-IF-PERSISTENT PIC S9(9) BINARY VALUE 4096.  350
COPIED        10 MQGMO-NO-SYNCPOINT            PIC S9(9) BINARY VALUE 4.     351
COPIED        10 MQGMO-MARK-SKIP-BACKOUT       PIC S9(9) BINARY VALUE 128.   352
COPIED        10 MQGMO-BROWSE-FIRST            PIC S9(9) BINARY VALUE 16.    353
COPIED        10 MQGMO-BROWSE-NEXT             PIC S9(9) BINARY VALUE 32.    354
COPIED        10 MQGMO-BROWSE-MSG-UNDER-CURSOR PIC S9(9) BINARY VALUE 2048.  355
COPIED        10 MQGMO-MSG-UNDER-CURSOR        PIC S9(9) BINARY VALUE 256.   356
COPIED        10 MQGMO-LOCK                    PIC S9(9) BINARY VALUE 512.   357
COPIED        10 MQGMO-UNLOCK                  PIC S9(9) BINARY VALUE 1024.  358
COPIED        10 MQGMO-ACCEPT-TRUNCATED-MSG    PIC S9(9) BINARY VALUE 64.    359
COPIED        10 MQGMO-CONVERT                 PIC S9(9) BINARY VALUE 16384. 360
COPIED        10 MQGMO-LOGICAL-ORDER           PIC S9(9) BINARY VALUE 32768. 361
COPIED        10 MQGMO-COMPLETE-MSG            PIC S9(9) BINARY VALUE 65536. 362
COPIED        10 MQGMO-ALL-MSGS-AVAILABLE      PIC S9(9) BINARY VALUE 131072.363
COPIED        10 MQGMO-ALL-SEGMENTS-AVAILABLE  PIC S9(9) BINARY VALUE 262144.364
COPIED        10 MQGMO-NONE                    PIC S9(9) BINARY VALUE 0.     365
COPIED                                                                          366
COPIED      **   Wait Interval                                                  367
COPIED        10 MQWI-UNLIMITED PIC S9(9) BINARY VALUE -1.                      368
COPIED                                                                          369
COPIED      **   Signal Values                                                  370
COPIED        10 MQEC-MSG-ARRIVED           PIC S9(9) BINARY VALUE 2.         371
COPIED        10 MQEC-WAIT-INTERVAL-EXPIRED PIC S9(9) BINARY VALUE 3.         372
COPIED        10 MQEC-WAIT-CANCELED         PIC S9(9) BINARY VALUE 4.         373
COPIED        10 MQEC-Q-MGR-QUIESCING       PIC S9(9) BINARY VALUE 5.         374
COPIED        10 MQEC-CONNECTION-QUIESCING  PIC S9(9) BINARY VALUE 6.         375
COPIED                                                                          376
COPIED      **   Match Options                                                  377
COPIED        10 MQMO-MATCH-MSG-ID         PIC S9(9) BINARY VALUE 1.          378
COPIED        10 MQMO-MATCH-CORREL-ID      PIC S9(9) BINARY VALUE 2.          379
COPIED        10 MQMO-MATCH-GROUP-ID       PIC S9(9) BINARY VALUE 4.          380
COPIED        10 MQMO-MATCH-MSG-SEQ-NUMBER PIC S9(9) BINARY VALUE 8.          381
COPIED        10 MQMO-MATCH-MSG-TOKEN      PIC S9(9) BINARY VALUE 32.         382
COPIED        10 MQMO-NONE                 PIC S9(9) BINARY VALUE 0.          383
COPIED                                                                          384
COPIED      **   Group Status                                                   385
COPIED        10 MQGS-NOT-IN-GROUP      PIC X VALUE ' '.                      386
COPIED        10 MQGS-MSG-IN-GROUP      PIC X VALUE 'G'.                      387
COPIED        10 MQGS-LAST-MSG-IN-GROUP PIC X VALUE 'L'.                      388
COPIED                                                                          389
COPIED      **   Segment Status                                                 390

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE     9
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQSS-NOT-A-SEGMENT PIC X VALUE ' '.                            391
COPIED        10 MQSS-SEGMENT       PIC X VALUE 'S'.                          392
COPIED        10 MQSS-LAST-SEGMENT  PIC X VALUE 'L'.                          393
COPIED                                                                          394
COPIED      **   Segmentation                                                   395
COPIED        10 MQSEG-INHIBITED PIC X VALUE ' '.                             396
COPIED        10 MQSEG-ALLOWED   PIC X VALUE 'A'.                             397
COPIED                                                                          398
COPIED      **   Message Token                                                  399
COPIED        10 MQMTOK-NONE PIC X(16) VALUE LOW-VALUES.                       400
COPIED                                                                          401
COPIED      **   Returned Length                                                402
COPIED        10 MQRL-UNDEFINED PIC S9(9) BINARY VALUE -1.                      403
COPIED                                                                          404
COPIED                                                                          405
COPIED      ****************************************************************    406
COPIED      **  Values Related to MQIIH Structure                         **    407
COPIED      ****************************************************************    408
```

Figure 23 MQPUT Sample Compiled Listing (Page 6 of 32)

```
COPIED                                                                          409
COPIED       **   Structure Identifier                                          410
COPIED         10 MQIIH-STRUC-ID PIC X(4) VALUE 'IIH '.                         411
COPIED                                                                          412
COPIED       **   Structure Version Number                                      413
COPIED         10 MQIIH-VERSION-1       PIC S9(9) BINARY VALUE 1.               414
COPIED         10 MQIIH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.               415
COPIED                                                                          416
COPIED       **   Structure Length                                              417
COPIED         10 MQIIH-LENGTH-1 PIC S9(9) BINARY VALUE 84.                     418
COPIED                                                                          419
COPIED       **   Flags                                                         420
COPIED         10 MQIIH-NONE PIC S9(9) BINARY VALUE 0.                          421
COPIED                                                                          422
COPIED       **   Authenticator                                                 423
COPIED         10 MQIAUT-NONE PIC X(8) VALUE SPACES.                            424
COPIED                                                                          425
COPIED       **   Transaction Instance Identifier                               426
COPIED         10 MQITII-NONE PIC X(16) VALUE LOW-VALUES.                       427
COPIED                                                                          428
COPIED       **   Transaction State                                             429
COPIED         10 MQITS-IN-CONVERSATION     PIC X VALUE 'C'.                    430
COPIED         10 MQITS-NOT-IN-CONVERSATION PIC X VALUE ' '.                    431
COPIED         10 MQITS-ARCHITECTED         PIC X VALUE 'A'.                    432
COPIED                                                                          433
COPIED       **   Commit Mode                                                   434
COPIED         10 MQICM-COMMIT-THEN-SEND PIC X VALUE '0'.                       435
COPIED         10 MQICM-SEND-THEN-COMMIT PIC X VALUE '1'.                       436
COPIED                                                                          437
COPIED       **   Security Scope                                                438
COPIED         10 MQISS-CHECK PIC X VALUE 'C'.                                  439

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    10
      1--------------------VISION:RESULTS FREE FORM TEXT----------------------72         --------
COPIED         10 MQISS-FULL  PIC X VALUE 'F'.                                  440
COPIED                                                                          441
COPIED                                                                          442
COPIED       *****************************************************************  443
COPIED       **  Values Related to MQMD Structure                           **   444
COPIED       *****************************************************************  445
COPIED                                                                          446
COPIED       **   Structure Identifier                                          447
COPIED         10 MQMD-STRUC-ID PIC X(4) VALUE 'MD  '.                          448
COPIED                                                                          449
COPIED       **   Structure Version Number                                      450
COPIED         10 MQMD-VERSION-1       PIC S9(9) BINARY VALUE 1.               451
COPIED         10 MQMD-VERSION-2       PIC S9(9) BINARY VALUE 2.               452
COPIED         10 MQMD-CURRENT-VERSION PIC S9(9) BINARY VALUE 2.               453
COPIED                                                                          454
COPIED       **   Report Options                                                455
COPIED         10 MQRO-EXCEPTION                PIC S9(9) BINARY VALUE         456
COPIED                                                          16777216.       457
COPIED         10 MQRO-EXCEPTION-WITH-DATA      PIC S9(9) BINARY VALUE         458
COPIED                                                          50331648.       459
COPIED         10 MQRO-EXCEPTION-WITH-FULL-DATA PIC S9(9) BINARY VALUE         460
COPIED                                                         117440512.       461
COPIED         10 MQRO-EXPIRATION               PIC S9(9) BINARY VALUE         462
COPIED                                                           2097152.       463
COPIED         10 MQRO-EXPIRATION-WITH-DATA     PIC S9(9) BINARY VALUE         464
COPIED                                                           6291456.       465
COPIED         10 MQRO-EXPIRATION-WITH-FULL-DATA PIC S9(9) BINARY VALUE        466
COPIED                                                          14680064.       467
COPIED         10 MQRO-COA                      PIC S9(9) BINARY VALUE 256.    468
COPIED         10 MQRO-COA-WITH-DATA            PIC S9(9) BINARY VALUE 768.    469
COPIED         10 MQRO-COA-WITH-FULL-DATA       PIC S9(9) BINARY VALUE 1792.   470
COPIED         10 MQRO-COD                      PIC S9(9) BINARY VALUE 2048.   471
COPIED         10 MQRO-COD-WITH-DATA            PIC S9(9) BINARY VALUE 6144.   472
COPIED         10 MQRO-COD-WITH-FULL-DATA       PIC S9(9) BINARY VALUE 14336.  473
COPIED         10 MQRO-PAN                      PIC S9(9) BINARY VALUE 1.      474
COPIED         10 MQRO-NAN                      PIC S9(9) BINARY VALUE 2.      475
COPIED         10 MQRO-NEW-MSG-ID               PIC S9(9) BINARY VALUE 0.      476
COPIED         10 MQRO-PASS-MSG-ID              PIC S9(9) BINARY VALUE 128.    477
COPIED         10 MQRO-COPY-MSG-ID-TO-CORREL-ID PIC S9(9) BINARY VALUE 0.      478
COPIED         10 MQRO-PASS-CORREL-ID           PIC S9(9) BINARY VALUE 64.     479
COPIED         10 MQRO-DEAD-LETTER-Q            PIC S9(9) BINARY VALUE 0.      480
COPIED         10 MQRO-DISCARD-MSG              PIC S9(9) BINARY VALUE         481
COPIED                                                         134217728.       482
COPIED         10 MQRO-NONE                     PIC S9(9) BINARY VALUE 0.      483
COPIED                                                                          484
COPIED       **   Report Options Masks                                          485
COPIED         10 MQRO-REJECT-UNSUP-MASK        PIC S9(9) BINARY VALUE         486
COPIED                                                         270270464.       487
COPIED         10 MQRO-ACCEPT-UNSUP-MASK        PIC S9(9) BINARY VALUE         488

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    11
```

Figure 23 MQPUT Sample Compiled Listing (Page 7 of 32)

```
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED                                                        -270532353.
COPIED         10 MQRO-ACCEPT-UNSUP-IF-XMIT-MASK PIC S9(9) BINARY VALUE 261888.
COPIED
COPIED       **  Message Types
COPIED         10 MQMT-SYSTEM-FIRST       PIC S9(9) BINARY VALUE 1.
COPIED         10 MQMT-REQUEST            PIC S9(9) BINARY VALUE 1.
COPIED         10 MQMT-REPLY              PIC S9(9) BINARY VALUE 2.
COPIED         10 MQMT-DATAGRAM           PIC S9(9) BINARY VALUE 8.
COPIED         10 MQMT-REPORT             PIC S9(9) BINARY VALUE 4.
COPIED         10 MQMT-MQE-FIELDS-FROM-MQE PIC S9(9) BINARY VALUE 112.
COPIED         10 MQMT-MQE-FIELDS         PIC S9(9) BINARY VALUE 113.
COPIED         10 MQMT-SYSTEM-LAST        PIC S9(9) BINARY VALUE 65535.
COPIED         10 MQMT-APPL-FIRST         PIC S9(9) BINARY VALUE 65536.
COPIED         10 MQMT-APPL-LAST          PIC S9(9) BINARY VALUE 999999999.
COPIED
COPIED       **  Expiry
COPIED         10 MQEI-UNLIMITED PIC S9(9) BINARY VALUE -1.
COPIED
COPIED       **  Feedback Values
COPIED         10 MQFB-NONE                  PIC S9(9) BINARY VALUE 0.
COPIED         10 MQFB-SYSTEM-FIRST          PIC S9(9) BINARY VALUE 1.
COPIED         10 MQFB-QUIT                  PIC S9(9) BINARY VALUE 256.
COPIED         10 MQFB-EXPIRATION            PIC S9(9) BINARY VALUE 258.
COPIED         10 MQFB-COA                   PIC S9(9) BINARY VALUE 259.
COPIED         10 MQFB-COD                   PIC S9(9) BINARY VALUE 260.
COPIED         10 MQFB-CHANNEL-COMPLETED     PIC S9(9) BINARY VALUE 262.
COPIED         10 MQFB-CHANNEL-FAIL-RETRY    PIC S9(9) BINARY VALUE 263.
COPIED         10 MQFB-CHANNEL-FAIL          PIC S9(9) BINARY VALUE 264.
COPIED         10 MQFB-APPL-CANNOT-BE-STARTED PIC S9(9) BINARY VALUE 265.
COPIED         10 MQFB-TM-ERROR              PIC S9(9) BINARY VALUE 266.
COPIED         10 MQFB-APPL-TYPE-ERROR       PIC S9(9) BINARY VALUE 267.
COPIED         10 MQFB-STOPPED-BY-MSG-EXIT   PIC S9(9) BINARY VALUE 268.
COPIED         10 MQFB-XMIT-Q-MSG-ERROR      PIC S9(9) BINARY VALUE 271.
COPIED         10 MQFB-PAN                   PIC S9(9) BINARY VALUE 275.
COPIED         10 MQFB-NAN                   PIC S9(9) BINARY VALUE 276.
COPIED         10 MQFB-STOPPED-BY-CHAD-EXIT  PIC S9(9) BINARY VALUE 277.
COPIED         10 MQFB-STOPPED-BY-PUBSUB-EXIT PIC S9(9) BINARY VALUE 279.
COPIED         10 MQFB-NOT-A-REPOSITORY-MSG  PIC S9(9) BINARY VALUE 280.
COPIED         10 MQFB-BIND-OPEN-CLUSRCVR-DEL PIC S9(9) BINARY VALUE 281.
COPIED         10 MQFB-DATA-LENGTH-ZERO      PIC S9(9) BINARY VALUE 291.
COPIED         10 MQFB-DATA-LENGTH-NEGATIVE  PIC S9(9) BINARY VALUE 292.
COPIED         10 MQFB-DATA-LENGTH-TOO-BIG   PIC S9(9) BINARY VALUE 293.
COPIED         10 MQFB-BUFFER-OVERFLOW       PIC S9(9) BINARY VALUE 294.
COPIED         10 MQFB-LENGTH-OFF-BY-ONE     PIC S9(9) BINARY VALUE 295.
COPIED         10 MQFB-IIH-ERROR             PIC S9(9) BINARY VALUE 296.
COPIED         10 MQFB-NOT-AUTHORIZED-FOR-IMS PIC S9(9) BINARY VALUE 298.
COPIED         10 MQFB-IMS-ERROR             PIC S9(9) BINARY VALUE 300.
COPIED         10 MQFB-IMS-FIRST             PIC S9(9) BINARY VALUE 301.
COPIED         10 MQFB-IMS-LAST              PIC S9(9) BINARY VALUE 399.

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    12
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED         10 MQFB-CICS-INTERNAL-ERROR   PIC S9(9) BINARY VALUE 401.
COPIED         10 MQFB-CICS-NOT-AUTHORIZED   PIC S9(9) BINARY VALUE 402.
COPIED         10 MQFB-CICS-BRIDGE-FAILURE   PIC S9(9) BINARY VALUE 403.
COPIED         10 MQFB-CICS-CORREL-ID-ERROR  PIC S9(9) BINARY VALUE 404.
COPIED         10 MQFB-CICS-CCSID-ERROR      PIC S9(9) BINARY VALUE 405.
COPIED         10 MQFB-CICS-ENCODING-ERROR   PIC S9(9) BINARY VALUE 406.
COPIED         10 MQFB-CICS-CIH-ERROR        PIC S9(9) BINARY VALUE 407.
COPIED         10 MQFB-CICS-UOW-ERROR        PIC S9(9) BINARY VALUE 408.
COPIED         10 MQFB-CICS-COMMAREA-ERROR   PIC S9(9) BINARY VALUE 409.
COPIED         10 MQFB-CICS-APPL-NOT-STARTED PIC S9(9) BINARY VALUE 410.
COPIED         10 MQFB-CICS-APPL-ABENDED     PIC S9(9) BINARY VALUE 411.
COPIED         10 MQFB-CICS-DLQ-ERROR        PIC S9(9) BINARY VALUE 412.
COPIED         10 MQFB-CICS-UOW-BACKED-OUT   PIC S9(9) BINARY VALUE 413.
COPIED         10 MQFB-SYSTEM-LAST           PIC S9(9) BINARY VALUE 65535.
COPIED         10 MQFB-APPL-FIRST            PIC S9(9) BINARY VALUE 65536.
COPIED         10 MQFB-APPL-LAST             PIC S9(9) BINARY VALUE 999999999.
COPIED
COPIED       **  Encoding
COPIED         10 MQENC-NATIVE PIC S9(9) BINARY VALUE 785.
COPIED
COPIED       **  Encoding Masks
COPIED         10 MQENC-INTEGER-MASK  PIC S9(9) BINARY VALUE 15.
COPIED         10 MQENC-DECIMAL-MASK  PIC S9(9) BINARY VALUE 240.
COPIED         10 MQENC-FLOAT-MASK    PIC S9(9) BINARY VALUE 3840.
COPIED         10 MQENC-RESERVED-MASK PIC S9(9) BINARY VALUE -4096.
COPIED
COPIED       **  Encodings for Binary Integers
COPIED         10 MQENC-INTEGER-UNDEFINED PIC S9(9) BINARY VALUE 0.
COPIED         10 MQENC-INTEGER-NORMAL    PIC S9(9) BINARY VALUE 1.
COPIED         10 MQENC-INTEGER-REVERSED  PIC S9(9) BINARY VALUE 2.
COPIED
COPIED       **  Encodings for Packed-Decimal Integers
```

Figure 23   MQPUT Sample Compiled Listing (Page 8 of 32)

```
COPIED        10 MQENC-DECIMAL-UNDEFINED PIC S9(9) BINARY VALUE 0.                    570
COPIED        10 MQENC-DECIMAL-NORMAL    PIC S9(9) BINARY VALUE 16.                   571
COPIED        10 MQENC-DECIMAL-REVERSED  PIC S9(9) BINARY VALUE 32.                   572
COPIED                                                                                573
COPIED      **   Encodings for Floating-Point Numbers                                 574
COPIED        10 MQENC-FLOAT-UNDEFINED     PIC S9(9) BINARY VALUE 0.                575
COPIED        10 MQENC-FLOAT-IEEE-NORMAL   PIC S9(9) BINARY VALUE 256.              576
COPIED        10 MQENC-FLOAT-IEEE-REVERSED PIC S9(9) BINARY VALUE 512.              577
COPIED        10 MQENC-FLOAT-S390          PIC S9(9) BINARY VALUE 768.              578
COPIED                                                                                579
COPIED      **   Coded Character-Set Identifiers                                      580
COPIED        10 MQCCSI-UNDEFINED PIC S9(9) BINARY VALUE 0.                          581
COPIED        10 MQCCSI-DEFAULT   PIC S9(9) BINARY VALUE 0.                          582
COPIED        10 MQCCSI-Q-MGR     PIC S9(9) BINARY VALUE 0.                          583
COPIED        10 MQCCSI-INHERIT   PIC S9(9) BINARY VALUE -2.                         584
COPIED        10 MQCCSI-EMBEDDED  PIC S9(9) BINARY VALUE -1.                         585
COPIED                                                                                586

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    13
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
COPIED      **   Formats                                                              587
COPIED        10 MQFMT-NONE              PIC X(8) VALUE SPACES.                     588
COPIED        10 MQFMT-ADMIN             PIC X(8) VALUE 'MQADMIN '.                 589
COPIED        10 MQFMT-CHANNEL-COMPLETED PIC X(8) VALUE 'MQCHCOM '.                 590
COPIED        10 MQFMT-CICS              PIC X(8) VALUE 'MQCICS  '.                 591
COPIED        10 MQFMT-COMMAND-1         PIC X(8) VALUE 'MQCMD1  '.                 592
COPIED        10 MQFMT-COMMAND-2         PIC X(8) VALUE 'MQCMD2  '.                 593
COPIED        10 MQFMT-DEAD-LETTER-HEADER PIC X(8) VALUE 'MQDEAD  '.                 594
COPIED        10 MQFMT-EVENT             PIC X(8) VALUE 'MQEVENT '.                 595
COPIED        10 MQFMT-IMS               PIC X(8) VALUE 'MQIMS   '.                 596
COPIED        10 MQFMT-IMS-VAR-STRING    PIC X(8) VALUE 'MQIMSVS '.                 597
COPIED        10 MQFMT-MD-EXTENSION      PIC X(8) VALUE 'MQHMDE  '.                 598
COPIED        10 MQFMT-PCF               PIC X(8) VALUE 'MQPCF   '.                 599
COPIED        10 MQFMT-REF-MSG-HEADER    PIC X(8) VALUE 'MQHREF  '.                 600
COPIED        10 MQFMT-RF-HEADER         PIC X(8) VALUE 'MQHRF   '.                 601
COPIED        10 MQFMT-RF-HEADER-2       PIC X(8) VALUE 'MQHRF2  '.                 602
COPIED        10 MQFMT-STRING            PIC X(8) VALUE 'MQSTR   '.                 603
COPIED        10 MQFMT-TRIGGER           PIC X(8) VALUE 'MQTRIG  '.                 604
COPIED        10 MQFMT-WORK-INFO-HEADER  PIC X(8) VALUE 'MQHWIH  '.                 605
COPIED        10 MQFMT-XMIT-Q-HEADER     PIC X(8) VALUE 'MQXMIT  '.                 606
COPIED                                                                                607
COPIED      **   Priority                                                             608
COPIED        10 MQPRI-PRIORITY-AS-Q-DEF PIC S9(9) BINARY VALUE -1.                  609
COPIED                                                                                610
COPIED      **   Persistence Values                                                   611
COPIED        10 MQPER-NOT-PERSISTENT      PIC S9(9) BINARY VALUE 0.                612
COPIED        10 MQPER-PERSISTENT          PIC S9(9) BINARY VALUE 1.                613
COPIED        10 MQPER-PERSISTENCE-AS-Q-DEF PIC S9(9) BINARY VALUE 2.               614
COPIED                                                                                615
COPIED      **   Message Identifier                                                   616
COPIED        10 MQMI-NONE PIC X(24) VALUE LOW-VALUES.                               617
COPIED                                                                                618
COPIED      **   Correlation Identifier                                               619
COPIED        10 MQCI-NONE        PIC X(24) VALUE LOW-VALUES.                        620
COPIED        10 MQCI-NEW-SESSION.                                                   621
COPIED           15 MQCI-NEW-SESSION1  PIC X(9) VALUE                                622
COPIED                X'414D51214B45575F53'.                                          623
COPIED           15 MQCI-NEW-SESSION2  PIC X(9) VALUE                                624
COPIED                X'455353494F4E5F434F'.                                          625
COPIED           15 MQCI-NEW-SESSION3  PIC X(6) VALUE                                626
COPIED                X'5252454C4944'.                                                627
COPIED                                                                                628
COPIED      **   Accounting Token                                                     629
COPIED        10 MQACT-NONE PIC X(32) VALUE LOW-VALUES.                              630
COPIED                                                                                631
COPIED      **   Put Application Type                                                 632
COPIED        10 MQAT-UNKNOWN          PIC S9(9) BINARY VALUE -1.                   633
COPIED        10 MQAT-NO-CONTEXT       PIC S9(9) BINARY VALUE 0.                    634
COPIED        10 MQAT-CICS             PIC S9(9) BINARY VALUE 1.                    635

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    14
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
COPIED        10 MQAT-MVS              PIC S9(9) BINARY VALUE 2.                    636
COPIED        10 MQAT-OS390            PIC S9(9) BINARY VALUE 2.                    637
COPIED        10 MQAT-ZOS              PIC S9(9) BINARY VALUE 2.                    638
COPIED        10 MQAT-IMS              PIC S9(9) BINARY VALUE 3.                    639
COPIED        10 MQAT-OS2              PIC S9(9) BINARY VALUE 4.                    640
COPIED        10 MQAT-DOS              PIC S9(9) BINARY VALUE 5.                    641
COPIED        10 MQAT-AIX              PIC S9(9) BINARY VALUE 6.                    642
COPIED        10 MQAT-UNIX             PIC S9(9) BINARY VALUE 6.                    643
COPIED        10 MQAT-QMGR             PIC S9(9) BINARY VALUE 7.                    644
COPIED        10 MQAT-OS400            PIC S9(9) BINARY VALUE 8.                    645
COPIED        10 MQAT-WINDOWS          PIC S9(9) BINARY VALUE 9.                    646
COPIED        10 MQAT-CICS-VSE         PIC S9(9) BINARY VALUE 10.                   647
COPIED        10 MQAT-WINDOWS-NT       PIC S9(9) BINARY VALUE 11.                   648
```

Figure 23 MQPUT Sample Compiled Listing (Page 9 of 32)

```
COPIED        10 MQAT-VMS              PIC S9(9) BINARY VALUE 12.                          649
COPIED        10 MQAT-GUARDIAN         PIC S9(9) BINARY VALUE 13.                          650
COPIED        10 MQAT-NSK              PIC S9(9) BINARY VALUE 13.                          651
COPIED        10 MQAT-VOS              PIC S9(9) BINARY VALUE 14.                          652
COPIED        10 MQAT-IMS-BRIDGE       PIC S9(9) BINARY VALUE 19.                          653
COPIED        10 MQAT-XCF              PIC S9(9) BINARY VALUE 20.                          654
COPIED        10 MQAT-CICS-BRIDGE      PIC S9(9) BINARY VALUE 21.                          655
COPIED        10 MQAT-NOTES-AGENT      PIC S9(9) BINARY VALUE 22.                          656
COPIED        10 MQAT-USER             PIC S9(9) BINARY VALUE 25.                          657
COPIED        10 MQAT-BROKER           PIC S9(9) BINARY VALUE 26.                          658
COPIED        10 MQAT-JAVA             PIC S9(9) BINARY VALUE 28.                          659
COPIED        10 MQAT-DQM              PIC S9(9) BINARY VALUE 29.                          660
COPIED        10 MQAT-CHANNEL-INITIATOR PIC S9(9) BINARY VALUE 30.                         661
COPIED        10 MQAT-DEFAULT          PIC S9(9) BINARY VALUE 2.                           662
COPIED        10 MQAT-USER-FIRST       PIC S9(9) BINARY VALUE 65536.                       663
COPIED        10 MQAT-USER-LAST        PIC S9(9) BINARY VALUE 999999999.                   664
COPIED                                                                                     665
COPIED      **   Group Identifier                                                          666
COPIED        10 MQGI-NONE PIC X(24) VALUE LOW-VALUES.                                    667
COPIED                                                                                     668
COPIED      **   Message Flags                                                             669
COPIED        10 MQMF-SEGMENTATION-INHIBITED PIC S9(9) BINARY VALUE 0.                    670
COPIED        10 MQMF-SEGMENTATION-ALLOWED   PIC S9(9) BINARY VALUE 1.                    671
COPIED        10 MQMF-MSG-IN-GROUP           PIC S9(9) BINARY VALUE 8.                    672
COPIED        10 MQMF-LAST-MSG-IN-GROUP      PIC S9(9) BINARY VALUE 16.                   673
COPIED        10 MQMF-SEGMENT                PIC S9(9) BINARY VALUE 2.                    674
COPIED        10 MQMF-LAST-SEGMENT           PIC S9(9) BINARY VALUE 4.                    675
COPIED        10 MQMF-NONE                   PIC S9(9) BINARY VALUE 0.                    676
COPIED                                                                                     677
COPIED      **   Message Flags Masks                                                       678
COPIED        10 MQMF-REJECT-UNSUP-MASK        PIC S9(9) BINARY VALUE 4095.              679
COPIED        10 MQMF-ACCEPT-UNSUP-MASK        PIC S9(9) BINARY VALUE                    680
COPIED                                                      -1048576.                      681
COPIED        10 MQMF-ACCEPT-UNSUP-IF-XMIT-MASK PIC S9(9) BINARY VALUE                     682
COPIED                                                      1044480.                       683
COPIED                                                                                     684

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    15
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED      **   Original Length                                                           685
COPIED        10 MQOL-UNDEFINED PIC S9(9) BINARY VALUE -1.                                686
COPIED                                                                                     687
COPIED                                                                                     688
COPIED      ******************************************************************           689
COPIED      **  Values Related to MQMDE Structure                           **           690
COPIED      ******************************************************************           691
COPIED                                                                                     692
COPIED      **   Structure Identifier                                                      693
COPIED        10 MQMDE-STRUC-ID PIC X(4) VALUE 'MDE '.                                    694
COPIED                                                                                     695
COPIED      **   Structure Version Number                                                  696
COPIED        10 MQMDE-VERSION-2       PIC S9(9) BINARY VALUE 2.                         697
COPIED        10 MQMDE-CURRENT-VERSION PIC S9(9) BINARY VALUE 2.                          698
COPIED                                                                                     699
COPIED      **   Structure Length                                                          700
COPIED        10 MQMDE-LENGTH-2 PIC S9(9) BINARY VALUE 72.                                701
COPIED                                                                                     702
COPIED      **   General Flags                                                             703
COPIED        10 MQMDEF-NONE PIC S9(9) BINARY VALUE 0.                                    704
COPIED                                                                                     705
COPIED                                                                                     706
COPIED      ******************************************************************           707
COPIED      **  Values Related to MQOD Structure                            **           708
COPIED      ******************************************************************           709
COPIED                                                                                     710
COPIED      **   Structure Identifier                                                      711
COPIED        10 MQOD-STRUC-ID PIC X(4) VALUE 'OD  '.                                     712
COPIED                                                                                     713
COPIED      **   Structure Version Number                                                  714
COPIED        10 MQOD-VERSION-1       PIC S9(9) BINARY VALUE 1.                          715
COPIED        10 MQOD-VERSION-2       PIC S9(9) BINARY VALUE 2.                          716
COPIED        10 MQOD-VERSION-3       PIC S9(9) BINARY VALUE 3.                          717
COPIED        10 MQOD-CURRENT-VERSION PIC S9(9) BINARY VALUE 3.                          718
COPIED                                                                                     719
COPIED      **   Structure Length                                                          720
COPIED        10 MQOD-CURRENT-LENGTH PIC S9(9) BINARY VALUE 336.                          721
COPIED                                                                                     722
COPIED      **   Object Types                                                              723
COPIED        10 MQOT-Q             PIC S9(9) BINARY VALUE 1.                            724
COPIED        10 MQOT-NAMELIST      PIC S9(9) BINARY VALUE 2.                            725
COPIED        10 MQOT-PROCESS       PIC S9(9) BINARY VALUE 3.                            726
COPIED        10 MQOT-STORAGE-CLASS PIC S9(9) BINARY VALUE 4.                            727
COPIED        10 MQOT-Q-MGR         PIC S9(9) BINARY VALUE 5.                            728
COPIED        10 MQOT-CHANNEL       PIC S9(9) BINARY VALUE 6.                            729
COPIED        10 MQOT-AUTH-INFO     PIC S9(9) BINARY VALUE 7.                            730
```

Figure 23 MQPUT Sample Compiled Listing (Page 10 of 32)

```
COPIED        10 MQOT-CF-STRUC      PIC S9(9) BINARY VALUE 10.                                  731
COPIED        10 MQOT-RESERVED-1    PIC S9(9) BINARY VALUE 999.                                 732
COPIED                                                                                          733

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    16
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED      **   Extended Object Types                                                          734
COPIED        10 MQOT-ALL               PIC S9(9) BINARY VALUE 1001.                          735
COPIED        10 MQOT-ALIAS-Q           PIC S9(9) BINARY VALUE 1002.                          736
COPIED        10 MQOT-MODEL-Q           PIC S9(9) BINARY VALUE 1003.                          737
COPIED        10 MQOT-LOCAL-Q           PIC S9(9) BINARY VALUE 1004.                          738
COPIED        10 MQOT-REMOTE-Q          PIC S9(9) BINARY VALUE 1005.                          739
COPIED        10 MQOT-SENDER-CHANNEL    PIC S9(9) BINARY VALUE 1007.                          740
COPIED        10 MQOT-SERVER-CHANNEL    PIC S9(9) BINARY VALUE 1008.                          741
COPIED        10 MQOT-REQUESTER-CHANNEL PIC S9(9) BINARY VALUE 1009.                          742
COPIED        10 MQOT-RECEIVER-CHANNEL  PIC S9(9) BINARY VALUE 1010.                          743
COPIED        10 MQOT-CURRENT-CHANNEL   PIC S9(9) BINARY VALUE 1011.                          744
COPIED        10 MQOT-SAVED-CHANNEL     PIC S9(9) BINARY VALUE 1012.                          745
COPIED        10 MQOT-SVRCONN-CHANNEL   PIC S9(9) BINARY VALUE 1013.                          746
COPIED        10 MQOT-CLNTCONN-CHANNEL  PIC S9(9) BINARY VALUE 1014.                          747
COPIED                                                                                          748
COPIED      **   Security Identifier                                                            749
COPIED        10 MQSID-NONE PIC X(40) VALUE LOW-VALUES.                                       750
COPIED                                                                                          751
COPIED      **   Security Identifier Type                                                       752
COPIED        10 MQSIDT-NONE           PIC X VALUE X'00'.                                     753
COPIED        10 MQSIDT-NT-SECURITY-ID PIC X VALUE X'01'.                                     754
COPIED                                                                                          755
COPIED                                                                                          756
COPIED      ******************************************************************              757
COPIED      **  Values Related to MQPMO Structure                           **              758
COPIED      ******************************************************************              759
COPIED                                                                                          760
COPIED      **   Structure Identifier                                                           761
COPIED        10 MQPMO-STRUC-ID PIC X(4) VALUE 'PMO '.                                        762
COPIED                                                                                          763
COPIED      **   Structure Version Number                                                       764
COPIED        10 MQPMO-VERSION-1       PIC S9(9) BINARY VALUE 1.                            765
COPIED        10 MQPMO-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.                            766
COPIED                                                                                          767
COPIED      **   Structure Length                                                               768
COPIED        10 MQPMO-CURRENT-LENGTH PIC S9(9) BINARY VALUE 128.                             769
COPIED                                                                                          770
COPIED      **   Put-Message Options                                                            771
COPIED        10 MQPMO-SYNCPOINT               PIC S9(9) BINARY VALUE 2.                    772
COPIED        10 MQPMO-NO-SYNCPOINT            PIC S9(9) BINARY VALUE 4.                    773
COPIED        10 MQPMO-LOGICAL-ORDER           PIC S9(9) BINARY VALUE 32768.                774
COPIED        10 MQPMO-NO-CONTEXT              PIC S9(9) BINARY VALUE 16384.                775
COPIED        10 MQPMO-DEFAULT-CONTEXT         PIC S9(9) BINARY VALUE 32.                   776
COPIED        10 MQPMO-PASS-IDENTITY-CONTEXT   PIC S9(9) BINARY VALUE 256.                  777
COPIED        10 MQPMO-PASS-ALL-CONTEXT        PIC S9(9) BINARY VALUE 512.                  778
COPIED        10 MQPMO-SET-IDENTITY-CONTEXT    PIC S9(9) BINARY VALUE 1024.                 779
COPIED        10 MQPMO-SET-ALL-CONTEXT         PIC S9(9) BINARY VALUE 2048.                 780
COPIED        10 MQPMO-ALTERNATE-USER-AUTHORITY PIC S9(9) BINARY VALUE 4096.                781
COPIED        10 MQPMO-FAIL-IF-QUIESCING       PIC S9(9) BINARY VALUE 8192.                 782

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    17
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQPMO-NONE                    PIC S9(9) BINARY VALUE 0.                    783
COPIED                                                                                          784
COPIED                                                                                          785
COPIED      ******************************************************************              786
COPIED      **  Values Related to MQRFH Structure                           **              787
COPIED      ******************************************************************              788
COPIED                                                                                          789
COPIED      **   Structure Identifier                                                           790
COPIED        10 MQRFH-STRUC-ID PIC X(4) VALUE 'RFH '.                                        791
COPIED                                                                                          792
COPIED      **   Structure Version Number                                                       793
COPIED        10 MQRFH-VERSION-1 PIC S9(9) BINARY VALUE 1.                                    794
COPIED        10 MQRFH-VERSION-2 PIC S9(9) BINARY VALUE 2.                                    795
COPIED                                                                                          796
COPIED      **   Structure Length                                                               797
COPIED        10 MQRFH-STRUC-LENGTH-FIXED   PIC S9(9) BINARY VALUE 32.                       798
COPIED        10 MQRFH-STRUC-LENGTH-FIXED-2 PIC S9(9) BINARY VALUE 36.                        799
COPIED                                                                                          800
COPIED      **   Flags                                                                          801
COPIED        10 MQRFH-NONE PIC S9(9) BINARY VALUE 0.                                         802
COPIED                                                                                          803
COPIED      **   Names for Name/Value String                                                    804
COPIED        10 MQNVS-APPL-TYPE PIC X(12) VALUE 'OPT_APP_GRP '.                              805
COPIED        10 MQNVS-MSG-TYPE  PIC X(13) VALUE 'OPT_MSG_TYPE '.                             806
COPIED                                                                                          807
COPIED                                                                                          808
COPIED      ******************************************************************              809
```

Figure 23 MQPUT Sample Compiled Listing (Page 11 of 32)

```
COPIED     ** Values Related to MQRMH Structure                          **
COPIED     *****************************************************************
COPIED
COPIED     **  Structure Identifier
COPIED       10 MQRMH-STRUC-ID PIC X(4) VALUE 'RMH '.
COPIED
COPIED     **  Structure Version Number
COPIED       10 MQRMH-VERSION-1       PIC S9(9) BINARY VALUE 1.
COPIED       10 MQRMH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.
COPIED
COPIED     **  Flags
COPIED       10 MQRMHF-LAST     PIC S9(9) BINARY VALUE 1.
COPIED       10 MQRMHF-NOT-LAST PIC S9(9) BINARY VALUE 0.
COPIED
COPIED     **  Object Instance Identifier
COPIED       10 MQOII-NONE PIC X(24) VALUE LOW-VALUES.
COPIED
COPIED
COPIED     *****************************************************************
COPIED     ** Values Related to MQTM Structure                           **
COPIED     *****************************************************************
COPIED

       COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *            DATE  09/20/05          PAGE   18
       1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED     **  Structure Identifier
COPIED       10 MQTM-STRUC-ID PIC X(4) VALUE 'TM  '.
COPIED
COPIED     **  Structure Version Number
COPIED       10 MQTM-VERSION-1       PIC S9(9) BINARY VALUE 1.
COPIED       10 MQTM-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.
COPIED
COPIED
COPIED     *****************************************************************
COPIED     ** Values Related to MQTMC2 Structure                         **
COPIED     *****************************************************************
COPIED
COPIED     **  Structure Identifier
COPIED       10 MQTMC-STRUC-ID PIC X(4) VALUE 'TMC '.
COPIED
COPIED     **  Structure Version Number
COPIED       10 MQTMC-VERSION-1       PIC X(4) VALUE '   1'.
COPIED       10 MQTMC-VERSION-2       PIC X(4) VALUE '   2'.
COPIED       10 MQTMC-CURRENT-VERSION PIC X(4) VALUE '   2'.
COPIED
COPIED
COPIED     *****************************************************************
COPIED     ** Values Related to MQWIH Structure                          **
COPIED     *****************************************************************
COPIED
COPIED     **  Structure Identifier
COPIED       10 MQWIH-STRUC-ID PIC X(4) VALUE 'WIH '.
COPIED
COPIED     **  Structure Version Number
COPIED       10 MQWIH-VERSION-1       PIC S9(9) BINARY VALUE 1.
COPIED       10 MQWIH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.
COPIED
COPIED     **  Structure Length
COPIED       10 MQWIH-LENGTH-1       PIC S9(9) BINARY VALUE 120.
COPIED       10 MQWIH-CURRENT-LENGTH PIC S9(9) BINARY VALUE 120.
COPIED
COPIED     **  Flags
COPIED       10 MQWIH-NONE PIC S9(9) BINARY VALUE 0.
COPIED
COPIED
COPIED     *****************************************************************
COPIED     ** Values Related to MQXQH Structure                          **
COPIED     *****************************************************************
COPIED
COPIED     **  Structure Identifier
COPIED       10 MQXQH-STRUC-ID PIC X(4) VALUE 'XQH '.
COPIED
COPIED     **  Structure Version Number
COPIED       10 MQXQH-VERSION-1       PIC S9(9) BINARY VALUE 1.

       COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *            DATE  09/20/05          PAGE   19
       1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED       10 MQXQH-CURRENT-VERSION PIC S9(9) BINARY VALUE 1.
COPIED
COPIED
COPIED     *****************************************************************
COPIED     ** Values Related to MQCLOSE Call                             **
COPIED     *****************************************************************
COPIED
COPIED     **  Close Options
```

Figure 23   MQPUT Sample Compiled Listing (Page 12 of 32)

```
COPIED        10 MQCO-NONE         PIC S9(9) BINARY VALUE 0.                                  889
COPIED        10 MQCO-DELETE       PIC S9(9) BINARY VALUE 1.                                  890
COPIED        10 MQCO-DELETE-PURGE PIC S9(9) BINARY VALUE 2.                                  891
COPIED                                                                                        892
COPIED                                                                                        893
COPIED      ****************************************************************                894
COPIED      **  Values Related to MQINQ Call                                **                895
COPIED      ****************************************************************                896
COPIED                                                                                        897
COPIED      **   Byte-Attribute Selectors                                                     898
COPIED        10 MQBA-FIRST PIC S9(9) BINARY VALUE 6001.                                    899
COPIED        10 MQBA-LAST  PIC S9(9) BINARY VALUE 8000.                                    900
COPIED                                                                                        901
COPIED      **   Character-Attribute Selectors                                              902
COPIED        10 MQCA-ALTERATION-DATE       PIC S9(9) BINARY VALUE 2027.                   903
COPIED        10 MQCA-ALTERATION-TIME       PIC S9(9) BINARY VALUE 2028.                   904
COPIED        10 MQCA-APPL-ID               PIC S9(9) BINARY VALUE 2001.                   905
COPIED        10 MQCA-AUTH-INFO-CONN-NAME   PIC S9(9) BINARY VALUE 2053.                   906
COPIED        10 MQCA-AUTH-INFO-DESC        PIC S9(9) BINARY VALUE 2046.                   907
COPIED        10 MQCA-AUTH-INFO-NAME        PIC S9(9) BINARY VALUE 2045.                   908
COPIED        10 MQCA-BACKOUT-REQ-Q-NAME    PIC S9(9) BINARY VALUE 2019.                   909
COPIED        10 MQCA-BASE-Q-NAME           PIC S9(9) BINARY VALUE 2002.                   910
COPIED        10 MQCA-CF-STRUC-DESC         PIC S9(9) BINARY VALUE 2052.                   911
COPIED        10 MQCA-CF-STRUC-NAME         PIC S9(9) BINARY VALUE 2039.                   912
COPIED        10 MQCA-CHANNEL-AUTO-DEF-EXIT PIC S9(9) BINARY VALUE 2026.                   913
COPIED        10 MQCA-CLUSTER-DATE          PIC S9(9) BINARY VALUE 2037.                   914
COPIED        10 MQCA-CLUSTER-NAME          PIC S9(9) BINARY VALUE 2029.                   915
COPIED        10 MQCA-CLUSTER-NAMELIST      PIC S9(9) BINARY VALUE 2030.                   916
COPIED        10 MQCA-CLUSTER-Q-MGR-NAME    PIC S9(9) BINARY VALUE 2031.                   917
COPIED        10 MQCA-CLUSTER-TIME          PIC S9(9) BINARY VALUE 2038.                   918
COPIED        10 MQCA-CLUSTER-WORKLOAD-DATA PIC S9(9) BINARY VALUE 2034.                   919
COPIED        10 MQCA-CLUSTER-WORKLOAD-EXIT PIC S9(9) BINARY VALUE 2033.                   920
COPIED        10 MQCA-COMMAND-INPUT-Q-NAME  PIC S9(9) BINARY VALUE 2003.                   921
COPIED        10 MQCA-CREATION-DATE         PIC S9(9) BINARY VALUE 2004.                   922
COPIED        10 MQCA-CREATION-TIME         PIC S9(9) BINARY VALUE 2005.                   923
COPIED        10 MQCA-DEAD-LETTER-Q-NAME    PIC S9(9) BINARY VALUE 2006.                   924
COPIED        10 MQCA-DEF-XMIT-Q-NAME       PIC S9(9) BINARY VALUE 2025.                   925
COPIED        10 MQCA-ENV-DATA              PIC S9(9) BINARY VALUE 2007.                   926
COPIED        10 MQCA-FIRST                 PIC S9(9) BINARY VALUE 2001.                   927
COPIED        10 MQCA-IGQ-USER-ID           PIC S9(9) BINARY VALUE 2041.                   928
COPIED        10 MQCA-INITIATION-Q-NAME     PIC S9(9) BINARY VALUE 2008.                   929

       COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    20
       1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQCA-LAST                  PIC S9(9) BINARY VALUE 4000.                   930
COPIED        10 MQCA-LAST-USED             PIC S9(9) BINARY VALUE 2053.                   931
COPIED        10 MQCA-LDAP-PASSWORD         PIC S9(9) BINARY VALUE 2048.                   932
COPIED        10 MQCA-LDAP-USER-NAME        PIC S9(9) BINARY VALUE 2047.                   933
COPIED        10 MQCA-NAMELIST-DESC         PIC S9(9) BINARY VALUE 2009.                   934
COPIED        10 MQCA-NAMELIST-NAME         PIC S9(9) BINARY VALUE 2010.                   935
COPIED        10 MQCA-NAMES                 PIC S9(9) BINARY VALUE 2020.                   936
COPIED        10 MQCA-PROCESS-DESC          PIC S9(9) BINARY VALUE 2011.                   937
COPIED        10 MQCA-PROCESS-NAME          PIC S9(9) BINARY VALUE 2012.                   938
COPIED        10 MQCA-Q-DESC                PIC S9(9) BINARY VALUE 2013.                   939
COPIED        10 MQCA-Q-MGR-DESC            PIC S9(9) BINARY VALUE 2014.                   940
COPIED        10 MQCA-Q-MGR-IDENTIFIER      PIC S9(9) BINARY VALUE 2032.                   941
COPIED        10 MQCA-Q-MGR-NAME            PIC S9(9) BINARY VALUE 2015.                   942
COPIED        10 MQCA-Q-NAME                PIC S9(9) BINARY VALUE 2016.                   943
COPIED        10 MQCA-QSG-NAME              PIC S9(9) BINARY VALUE 2040.                   944
COPIED        10 MQCA-REMOTE-Q-MGR-NAME     PIC S9(9) BINARY VALUE 2017.                   945
COPIED        10 MQCA-REMOTE-Q-NAME         PIC S9(9) BINARY VALUE 2018.                   946
COPIED        10 MQCA-REPOSITORY-NAME       PIC S9(9) BINARY VALUE 2035.                   947
COPIED        10 MQCA-REPOSITORY-NAMELIST   PIC S9(9) BINARY VALUE 2036.                   948
COPIED        10 MQCA-SSL-CRL-NAMELIST      PIC S9(9) BINARY VALUE 2050.                   949
COPIED        10 MQCA-SSL-CRYPTO-HARDWARE   PIC S9(9) BINARY VALUE 2051.                   950
COPIED        10 MQCA-SSL-KEY-REPOSITORY    PIC S9(9) BINARY VALUE 2049.                   951
COPIED        10 MQCA-STORAGE-CLASS         PIC S9(9) BINARY VALUE 2022.                   952
COPIED        10 MQCA-STORAGE-CLASS-DESC    PIC S9(9) BINARY VALUE 2042.                   953
COPIED        10 MQCA-TRIGGER-DATA          PIC S9(9) BINARY VALUE 2023.                   954
COPIED        10 MQCA-USER-DATA             PIC S9(9) BINARY VALUE 2021.                   955
COPIED        10 MQCA-USER-LIST             PIC S9(9) BINARY VALUE 4000.                   956
COPIED        10 MQCA-XCF-GROUP-NAME        PIC S9(9) BINARY VALUE 2043.                   957
COPIED        10 MQCA-XCF-MEMBER-NAME       PIC S9(9) BINARY VALUE 2044.                   958
COPIED        10 MQCA-XMIT-Q-NAME           PIC S9(9) BINARY VALUE 2024.                   959
COPIED                                                                                       960
COPIED      **   Integer-Attribute Selectors                                                961
COPIED        10 MQIA-APPL-TYPE               PIC S9(9) BINARY VALUE 1.                    962
COPIED        10 MQIA-ARCHIVE                 PIC S9(9) BINARY VALUE 60.                   963
COPIED        10 MQIA-AUTH-INFO-TYPE          PIC S9(9) BINARY VALUE 66.                   964
COPIED        10 MQIA-AUTHORITY-EVENT         PIC S9(9) BINARY VALUE 47.                   965
COPIED        10 MQIA-BACKOUT-THRESHOLD       PIC S9(9) BINARY VALUE 22.                   966
COPIED        10 MQIA-CF-LEVEL                PIC S9(9) BINARY VALUE 70.                   967
COPIED        10 MQIA-CF-RECOVER              PIC S9(9) BINARY VALUE 71.                   968
COPIED        10 MQIA-CHANNEL-AUTO-DEF        PIC S9(9) BINARY VALUE 55.                   969
COPIED        10 MQIA-CHANNEL-AUTO-DEF-EVENT  PIC S9(9) BINARY VALUE 56.                   970
```

Figure 23 MQPUT Sample Compiled Listing (Page 13 of 32)

```
COPIED         10 MQIA-CLUSTER-Q-TYPE          PIC S9(9) BINARY VALUE 59.                                  971
COPIED         10 MQIA-CLUSTER-WORKLOAD-LENGTH  PIC S9(9) BINARY VALUE 58.                                  972
COPIED         10 MQIA-CODED-CHAR-SET-ID        PIC S9(9) BINARY VALUE 2.                                   973
COPIED         10 MQIA-COMMAND-LEVEL            PIC S9(9) BINARY VALUE 31.                                  974
COPIED         10 MQIA-CONFIGURATION-EVENT      PIC S9(9) BINARY VALUE 51.                                  975
COPIED         10 MQIA-CPI-LEVEL                PIC S9(9) BINARY VALUE 27.                                  976
COPIED         10 MQIA-CURRENT-Q-DEPTH          PIC S9(9) BINARY VALUE 3.                                   977
COPIED         10 MQIA-DEF-BIND                 PIC S9(9) BINARY VALUE 61.                                  978

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    21
       1--------------------VISION:RESULTS FREE FORM TEXT---------------------72          --------
COPIED         10 MQIA-DEF-INPUT-OPEN-OPTION    PIC S9(9) BINARY VALUE 4.                                   979
COPIED         10 MQIA-DEF-PERSISTENCE          PIC S9(9) BINARY VALUE 5.                                   980
COPIED         10 MQIA-DEF-PRIORITY             PIC S9(9) BINARY VALUE 6.                                   981
COPIED         10 MQIA-DEFINITION-TYPE          PIC S9(9) BINARY VALUE 7.                                   982
COPIED         10 MQIA-DIST-LISTS               PIC S9(9) BINARY VALUE 34.                                  983
COPIED         10 MQIA-EXPIRY-INTERVAL          PIC S9(9) BINARY VALUE 39.                                  984
COPIED         10 MQIA-FIRST                    PIC S9(9) BINARY VALUE 1.                                   985
COPIED         10 MQIA-HARDEN-GET-BACKOUT       PIC S9(9) BINARY VALUE 8.                                   986
COPIED         10 MQIA-HIGH-Q-DEPTH             PIC S9(9) BINARY VALUE 36.                                  987
COPIED         10 MQIA-IGQ-PUT-AUTHORITY        PIC S9(9) BINARY VALUE 65.                                  988
COPIED         10 MQIA-INDEX-TYPE               PIC S9(9) BINARY VALUE 57.                                  989
COPIED         10 MQIA-INHIBIT-EVENT            PIC S9(9) BINARY VALUE 48.                                  990
COPIED         10 MQIA-INHIBIT-GET              PIC S9(9) BINARY VALUE 9.                                   991
COPIED         10 MQIA-INHIBIT-PUT              PIC S9(9) BINARY VALUE 10.                                  992
COPIED         10 MQIA-INTRA-GROUP-QUEUING      PIC S9(9) BINARY VALUE 64.                                  993
COPIED         10 MQIA-LAST                     PIC S9(9) BINARY VALUE 2000.                                994
COPIED         10 MQIA-LAST-USED                PIC S9(9) BINARY VALUE 72.                                  995
COPIED         10 MQIA-LOCAL-EVENT              PIC S9(9) BINARY VALUE 49.                                  996
COPIED         10 MQIA-MAX-HANDLES              PIC S9(9) BINARY VALUE 11.                                  997
COPIED         10 MQIA-MAX-MSG-LENGTH           PIC S9(9) BINARY VALUE 13.                                  998
COPIED         10 MQIA-MAX-PRIORITY             PIC S9(9) BINARY VALUE 14.                                  999
COPIED         10 MQIA-MAX-Q-DEPTH              PIC S9(9) BINARY VALUE 15.                                 1000
COPIED         10 MQIA-MAX-UNCOMMITTED-MSGS     PIC S9(9) BINARY VALUE 33.                                 1001
COPIED         10 MQIA-MSG-DELIVERY-SEQUENCE    PIC S9(9) BINARY VALUE 16.                                 1002
COPIED         10 MQIA-MSG-DEQ-COUNT            PIC S9(9) BINARY VALUE 38.                                 1003
COPIED         10 MQIA-MSG-ENQ-COUNT            PIC S9(9) BINARY VALUE 37.                                 1004
COPIED         10 MQIA-NAME-COUNT               PIC S9(9) BINARY VALUE 19.                                 1005
COPIED         10 MQIA-NAMELIST-TYPE            PIC S9(9) BINARY VALUE 72.                                 1006
COPIED         10 MQIA-OPEN-INPUT-COUNT         PIC S9(9) BINARY VALUE 17.                                 1007
COPIED         10 MQIA-OPEN-OUTPUT-COUNT        PIC S9(9) BINARY VALUE 18.                                 1008
COPIED         10 MQIA-PAGESET-ID               PIC S9(9) BINARY VALUE 62.                                 1009
COPIED         10 MQIA-PERFORMANCE-EVENT        PIC S9(9) BINARY VALUE 53.                                 1010
COPIED         10 MQIA-PLATFORM                 PIC S9(9) BINARY VALUE 32.                                 1011
COPIED         10 MQIA-Q-DEPTH-HIGH-EVENT       PIC S9(9) BINARY VALUE 43.                                 1012
COPIED         10 MQIA-Q-DEPTH-HIGH-LIMIT       PIC S9(9) BINARY VALUE 40.                                 1013
COPIED         10 MQIA-Q-DEPTH-LOW-EVENT        PIC S9(9) BINARY VALUE 44.                                 1014
COPIED         10 MQIA-Q-DEPTH-LOW-LIMIT        PIC S9(9) BINARY VALUE 41.                                 1015
COPIED         10 MQIA-Q-DEPTH-MAX-EVENT        PIC S9(9) BINARY VALUE 42.                                 1016
COPIED         10 MQIA-Q-SERVICE-INTERVAL       PIC S9(9) BINARY VALUE 54.                                 1017
COPIED         10 MQIA-Q-SERVICE-INTERVAL-EVENT PIC S9(9) BINARY VALUE 46.                                 1018
COPIED         10 MQIA-Q-TYPE                   PIC S9(9) BINARY VALUE 20.                                 1019
COPIED         10 MQIA-QSG-DISP                 PIC S9(9) BINARY VALUE 63.                                 1020
COPIED         10 MQIA-REMOTE-EVENT             PIC S9(9) BINARY VALUE 50.                                 1021
COPIED         10 MQIA-RETENTION-INTERVAL       PIC S9(9) BINARY VALUE 21.                                 1022
COPIED         10 MQIA-SCOPE                    PIC S9(9) BINARY VALUE 45.                                 1023
COPIED         10 MQIA-SHAREABILITY             PIC S9(9) BINARY VALUE 23.                                 1024
COPIED         10 MQIA-SSL-TASKS                PIC S9(9) BINARY VALUE 69.                                 1025
COPIED         10 MQIA-START-STOP-EVENT         PIC S9(9) BINARY VALUE 52.                                 1026
COPIED         10 MQIA-SYNCPOINT                PIC S9(9) BINARY VALUE 30.                                 1027

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    22
       1--------------------VISION:RESULTS FREE FORM TEXT---------------------72          --------
COPIED         10 MQIA-TIME-SINCE-RESET         PIC S9(9) BINARY VALUE 35.                                 1028
COPIED         10 MQIA-TRIGGER-CONTROL          PIC S9(9) BINARY VALUE 24.                                 1029
COPIED         10 MQIA-TRIGGER-DEPTH            PIC S9(9) BINARY VALUE 29.                                 1030
COPIED         10 MQIA-TRIGGER-INTERVAL         PIC S9(9) BINARY VALUE 25.                                 1031
COPIED         10 MQIA-TRIGGER-MSG-PRIORITY     PIC S9(9) BINARY VALUE 26.                                 1032
COPIED         10 MQIA-TRIGGER-TYPE             PIC S9(9) BINARY VALUE 28.                                 1033
COPIED         10 MQIA-USAGE                    PIC S9(9) BINARY VALUE 12.                                 1034
COPIED         10 MQIA-USER-LIST                PIC S9(9) BINARY VALUE 2000.                               1035
COPIED                                                                                                     1036
COPIED       **   Integer Attribute Value Denoting "Not Applicable"                                        1037
COPIED         10 MQIAV-NOT-APPLICABLE PIC S9(9) BINARY VALUE -1.                                         1038
COPIED         10 MQIAV-UNDEFINED      PIC S9(9) BINARY VALUE -2.                                         1039
COPIED                                                                                                     1040
COPIED                                                                                                     1041
COPIED       ****************************************************************                            1042
COPIED       **  Values Related to MQOPEN Call                              **                            1043
COPIED       ****************************************************************                            1044
COPIED                                                                                                     1045
COPIED       **   Open Options                                                                            1046
COPIED         10 MQOO-INPUT-AS-Q-DEF           PIC S9(9) BINARY VALUE 1.                                 1047
COPIED         10 MQOO-INPUT-SHARED             PIC S9(9) BINARY VALUE 2.                                 1048
COPIED         10 MQOO-INPUT-EXCLUSIVE          PIC S9(9) BINARY VALUE 4.                                 1049
```

Figure 23 MQPUT Sample Compiled Listing (Page 14 of 32)

```
COPIED        10 MQOO-BROWSE                 PIC S9(9) BINARY VALUE 8.                                1050
COPIED        10 MQOO-OUTPUT                 PIC S9(9) BINARY VALUE 16.                               1051
COPIED        10 MQOO-INQUIRE                PIC S9(9) BINARY VALUE 32.                               1052
COPIED        10 MQOO-SET                    PIC S9(9) BINARY VALUE 64.                               1053
COPIED        10 MQOO-BIND-ON-OPEN           PIC S9(9) BINARY VALUE 16384.                            1054
COPIED        10 MQOO-BIND-NOT-FIXED         PIC S9(9) BINARY VALUE 32768.                            1055
COPIED        10 MQOO-BIND-AS-Q-DEF          PIC S9(9) BINARY VALUE 0.                                1056
COPIED        10 MQOO-SAVE-ALL-CONTEXT       PIC S9(9) BINARY VALUE 128.                              1057
COPIED        10 MQOO-PASS-IDENTITY-CONTEXT  PIC S9(9) BINARY VALUE 256.                              1058
COPIED        10 MQOO-PASS-ALL-CONTEXT       PIC S9(9) BINARY VALUE 512.                              1059
COPIED        10 MQOO-SET-IDENTITY-CONTEXT   PIC S9(9) BINARY VALUE 1024.                             1060
COPIED        10 MQOO-SET-ALL-CONTEXT        PIC S9(9) BINARY VALUE 2048.                             1061
COPIED        10 MQOO-ALTERNATE-USER-AUTHORITY PIC S9(9) BINARY VALUE 4096.                           1062
COPIED        10 MQOO-FAIL-IF-QUIESCING      PIC S9(9) BINARY VALUE 8192.                             1063
COPIED                                                                                                1064
COPIED                                                                                                1065
COPIED      ******************************************************************                        1066
COPIED      **  Values Related to All Calls                                 **                        1067
COPIED      ******************************************************************                        1068
COPIED                                                                                                1069
COPIED      **   Connection Handle                                                                    1070
COPIED        10 MQHC-DEF-HCONN PIC S9(9) BINARY VALUE 0.                                             1071
COPIED                                                                                                1072
COPIED      **   String Lengths                                                                       1073
COPIED        10 MQ-ABEND-CODE-LENGTH        PIC S9(9) BINARY VALUE 4.                                1074
COPIED        10 MQ-ACCOUNTING-TOKEN-LENGTH  PIC S9(9) BINARY VALUE 32.                               1075
COPIED        10 MQ-APPL-IDENTITY-DATA-LENGTH PIC S9(9) BINARY VALUE 32.                              1076

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    23
       1------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQ-APPL-NAME-LENGTH         PIC S9(9) BINARY VALUE 28.                               1077
COPIED        10 MQ-APPL-ORIGIN-DATA-LENGTH  PIC S9(9) BINARY VALUE 4.                                1078
COPIED        10 MQ-APPL-TAG-LENGTH          PIC S9(9) BINARY VALUE 28.                               1079
COPIED        10 MQ-ATTENTION-ID-LENGTH      PIC S9(9) BINARY VALUE 4.                                1080
COPIED        10 MQ-AUTH-INFO-CONN-NAME-LENGTH PIC S9(9) BINARY VALUE 264.                            1081
COPIED        10 MQ-AUTH-INFO-DESC-LENGTH    PIC S9(9) BINARY VALUE 64.                               1082
COPIED        10 MQ-AUTH-INFO-NAME-LENGTH    PIC S9(9) BINARY VALUE 48.                               1083
COPIED        10 MQ-AUTHENTICATOR-LENGTH     PIC S9(9) BINARY VALUE 8.                                1084
COPIED        10 MQ-BRIDGE-NAME-LENGTH       PIC S9(9) BINARY VALUE 24.                               1085
COPIED        10 MQ-CANCEL-CODE-LENGTH       PIC S9(9) BINARY VALUE 4.                                1086
COPIED        10 MQ-CF-STRUC-DESC-LENGTH     PIC S9(9) BINARY VALUE 64.                               1087
COPIED        10 MQ-CF-STRUC-NAME-LENGTH     PIC S9(9) BINARY VALUE 12.                               1088
COPIED        10 MQ-CHANNEL-DATE-LENGTH      PIC S9(9) BINARY VALUE 12.                               1089
COPIED        10 MQ-CHANNEL-DESC-LENGTH      PIC S9(9) BINARY VALUE 64.                               1090
COPIED        10 MQ-CHANNEL-NAME-LENGTH      PIC S9(9) BINARY VALUE 20.                               1091
COPIED        10 MQ-CHANNEL-TIME-LENGTH      PIC S9(9) BINARY VALUE 8.                                1092
COPIED        10 MQ-CLUSTER-NAME-LENGTH      PIC S9(9) BINARY VALUE 48.                               1093
COPIED        10 MQ-CONN-NAME-LENGTH         PIC S9(9) BINARY VALUE 264.                              1094
COPIED        10 MQ-CONN-TAG-LENGTH          PIC S9(9) BINARY VALUE 128.                              1095
COPIED        10 MQ-CORREL-ID-LENGTH         PIC S9(9) BINARY VALUE 24.                               1096
COPIED        10 MQ-CREATION-DATE-LENGTH     PIC S9(9) BINARY VALUE 12.                               1097
COPIED        10 MQ-CREATION-TIME-LENGTH     PIC S9(9) BINARY VALUE 8.                                1098
COPIED        10 MQ-DATE-LENGTH              PIC S9(9) BINARY VALUE 12.                               1099
COPIED        10 MQ-DISTINGUISHED-NAME-LENGTH PIC S9(9) BINARY VALUE 1024.                            1100
COPIED        10 MQ-EXIT-DATA-LENGTH         PIC S9(9) BINARY VALUE 32.                               1101
COPIED        10 MQ-EXIT-NAME-LENGTH         PIC S9(9) BINARY VALUE 8.                                1102
COPIED        10 MQ-EXIT-USER-AREA-LENGTH    PIC S9(9) BINARY VALUE 16.                               1103
COPIED        10 MQ-FACILITY-LENGTH          PIC S9(9) BINARY VALUE 8.                                1104
COPIED        10 MQ-FACILITY-LIKE-LENGTH     PIC S9(9) BINARY VALUE 4.                                1105
COPIED        10 MQ-FORMAT-LENGTH            PIC S9(9) BINARY VALUE 8.                                1106
COPIED        10 MQ-FUNCTION-LENGTH          PIC S9(9) BINARY VALUE 4.                                1107
COPIED        10 MQ-GROUP-ID-LENGTH          PIC S9(9) BINARY VALUE 24.                               1108
COPIED        10 MQ-LDAP-PASSWORD-LENGTH     PIC S9(9) BINARY VALUE 32.                               1109
COPIED        10 MQ-LOCAL-ADDRESS-LENGTH     PIC S9(9) BINARY VALUE 48.                               1110
COPIED        10 MQ-LTERM-OVERRIDE-LENGTH    PIC S9(9) BINARY VALUE 8.                                1111
COPIED        10 MQ-LUWID-LENGTH             PIC S9(9) BINARY VALUE 16.                               1112
COPIED        10 MQ-MAX-EXIT-NAME-LENGTH     PIC S9(9) BINARY VALUE 128.                              1113
COPIED        10 MQ-MAX-MCA-USER-ID-LENGTH   PIC S9(9) BINARY VALUE 64.                               1114
COPIED        10 MQ-MAX-USER-ID-LENGTH       PIC S9(9) BINARY VALUE 64.                               1115
COPIED        10 MQ-MCA-JOB-NAME-LENGTH      PIC S9(9) BINARY VALUE 28.                               1116
COPIED        10 MQ-MCA-NAME-LENGTH          PIC S9(9) BINARY VALUE 20.                               1117
COPIED        10 MQ-MCA-USER-DATA-LENGTH     PIC S9(9) BINARY VALUE 32.                               1118
COPIED        10 MQ-MCA-USER-ID-LENGTH       PIC S9(9) BINARY VALUE 12.                               1119
COPIED        10 MQ-MFS-MAP-NAME-LENGTH      PIC S9(9) BINARY VALUE 8.                                1120
COPIED        10 MQ-MODE-NAME-LENGTH         PIC S9(9) BINARY VALUE 8.                                1121
COPIED        10 MQ-MSG-HEADER-LENGTH        PIC S9(9) BINARY VALUE 4000.                             1122
COPIED        10 MQ-MSG-ID-LENGTH            PIC S9(9) BINARY VALUE 24.                               1123
COPIED        10 MQ-MSG-TOKEN-LENGTH         PIC S9(9) BINARY VALUE 16.                               1124
COPIED        10 MQ-NAMELIST-DESC-LENGTH     PIC S9(9) BINARY VALUE 64.                               1125

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *             DATE  09/20/05            PAGE    24
       1------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQ-NAMELIST-NAME-LENGTH     PIC S9(9) BINARY VALUE 48.                               1126
COPIED        10 MQ-OBJECT-INSTANCE-ID-LENGTH PIC S9(9) BINARY VALUE 24.                              1127
COPIED        10 MQ-OBJECT-NAME-LENGTH       PIC S9(9) BINARY VALUE 48.                               1128
```

Figure 23 MQPUT Sample Compiled Listing (Page 15 of 32)

```
COPIED        10 MQ-PASSWORD-LENGTH            PIC S9(9) BINARY VALUE 12.                    1129
COPIED        10 MQ-PROCESS-APPL-ID-LENGTH     PIC S9(9) BINARY VALUE 256.                   1130
COPIED        10 MQ-PROCESS-DESC-LENGTH        PIC S9(9) BINARY VALUE 64.                    1131
COPIED        10 MQ-PROCESS-ENV-DATA-LENGTH    PIC S9(9) BINARY VALUE 128.                   1132
COPIED        10 MQ-PROCESS-NAME-LENGTH        PIC S9(9) BINARY VALUE 48.                    1133
COPIED        10 MQ-PROCESS-USER-DATA-LENGTH   PIC S9(9) BINARY VALUE 128.                   1134
COPIED        10 MQ-PROGRAM-NAME-LENGTH        PIC S9(9) BINARY VALUE 20.                    1135
COPIED        10 MQ-PUT-APPL-NAME-LENGTH       PIC S9(9) BINARY VALUE 28.                    1136
COPIED        10 MQ-PUT-DATE-LENGTH            PIC S9(9) BINARY VALUE 8.                     1137
COPIED        10 MQ-PUT-TIME-LENGTH            PIC S9(9) BINARY VALUE 8.                     1138
COPIED        10 MQ-Q-DESC-LENGTH              PIC S9(9) BINARY VALUE 64.                    1139
COPIED        10 MQ-Q-MGR-DESC-LENGTH          PIC S9(9) BINARY VALUE 64.                    1140
COPIED        10 MQ-Q-MGR-IDENTIFIER-LENGTH    PIC S9(9) BINARY VALUE 48.                    1141
COPIED        10 MQ-Q-MGR-NAME-LENGTH          PIC S9(9) BINARY VALUE 48.                    1142
COPIED        10 MQ-Q-NAME-LENGTH              PIC S9(9) BINARY VALUE 48.                    1143
COPIED        10 MQ-QSG-NAME-LENGTH            PIC S9(9) BINARY VALUE 4.                     1144
COPIED        10 MQ-REMOTE-SYS-ID-LENGTH       PIC S9(9) BINARY VALUE 4.                     1145
COPIED        10 MQ-SECURITY-ID-LENGTH         PIC S9(9) BINARY VALUE 40.                    1146
COPIED        10 MQ-SERVICE-NAME-LENGTH        PIC S9(9) BINARY VALUE 32.                    1147
COPIED        10 MQ-SERVICE-STEP-LENGTH        PIC S9(9) BINARY VALUE 8.                     1148
COPIED        10 MQ-SHORT-CONN-NAME-LENGTH     PIC S9(9) BINARY VALUE 20.                    1149
COPIED        10 MQ-SSL-CIPHER-SPEC-LENGTH     PIC S9(9) BINARY VALUE 32.                    1150
COPIED        10 MQ-SSL-CRYPTO-HARDWARE-LENGTH PIC S9(9) BINARY VALUE 256.                   1151
COPIED        10 MQ-SSL-HANDSHAKE-STAGE-LENGTH PIC S9(9) BINARY VALUE 32.                    1152
COPIED        10 MQ-SSL-KEY-REPOSITORY-LENGTH  PIC S9(9) BINARY VALUE 256.                   1153
COPIED        10 MQ-SSL-PEER-NAME-LENGTH       PIC S9(9) BINARY VALUE 1024.                  1154
COPIED        10 MQ-SSL-SHORT-PEER-NAME-LENGTH PIC S9(9) BINARY VALUE 256.                   1155
COPIED        10 MQ-START-CODE-LENGTH          PIC S9(9) BINARY VALUE 4.                     1156
COPIED        10 MQ-STORAGE-CLASS-DESC-LENGTH  PIC S9(9) BINARY VALUE 64.                    1157
COPIED        10 MQ-STORAGE-CLASS-LENGTH       PIC S9(9) BINARY VALUE 8.                     1158
COPIED        10 MQ-SUB-IDENTITY-LENGTH        PIC S9(9) BINARY VALUE 128.                   1159
COPIED        10 MQ-TIME-LENGTH                PIC S9(9) BINARY VALUE 8.                     1160
COPIED        10 MQ-TOTAL-EXIT-DATA-LENGTH     PIC S9(9) BINARY VALUE 999.                   1161
COPIED        10 MQ-TOTAL-EXIT-NAME-LENGTH     PIC S9(9) BINARY VALUE 999.                   1162
COPIED        10 MQ-TP-NAME-LENGTH             PIC S9(9) BINARY VALUE 64.                    1163
COPIED        10 MQ-TRAN-INSTANCE-ID-LENGTH    PIC S9(9) BINARY VALUE 16.                    1164
COPIED        10 MQ-TRANSACTION-ID-LENGTH      PIC S9(9) BINARY VALUE 4.                     1165
COPIED        10 MQ-TRIGGER-DATA-LENGTH        PIC S9(9) BINARY VALUE 64.                    1166
COPIED        10 MQ-USER-ID-LENGTH             PIC S9(9) BINARY VALUE 12.                    1167
COPIED        10 MQ-XCF-GROUP-NAME-LENGTH      PIC S9(9) BINARY VALUE 8.                     1168
COPIED        10 MQ-XCF-MEMBER-NAME-LENGTH     PIC S9(9) BINARY VALUE 16.                    1169
COPIED                                                                                       1170
COPIED      **  Completion Codes                                                             1171
COPIED        10 MQCC-OK      PIC S9(9) BINARY VALUE 0.                                      1172
COPIED        10 MQCC-WARNING PIC S9(9) BINARY VALUE 1.                                      1173
COPIED        10 MQCC-FAILED  PIC S9(9) BINARY VALUE 2.                                      1174

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    25
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED        10 MQCC-UNKNOWN PIC S9(9) BINARY VALUE -1.                                     1175
COPIED                                                                                       1176
COPIED      **  Reason Codes                                                                 1177
COPIED        10 MQRC-NONE                     PIC S9(9) BINARY VALUE 0.                     1178
COPIED        10 MQRC-APPL-FIRST               PIC S9(9) BINARY VALUE 900.                   1179
COPIED        10 MQRC-APPL-LAST                PIC S9(9) BINARY VALUE 999.                   1180
COPIED        10 MQRC-ALIAS-BASE-Q-TYPE-ERROR  PIC S9(9) BINARY VALUE 2001.                  1181
COPIED        10 MQRC-ALREADY-CONNECTED        PIC S9(9) BINARY VALUE 2002.                  1182
COPIED        10 MQRC-BACKED-OUT               PIC S9(9) BINARY VALUE 2003.                  1183
COPIED        10 MQRC-BUFFER-ERROR             PIC S9(9) BINARY VALUE 2004.                  1184
COPIED        10 MQRC-BUFFER-LENGTH-ERROR      PIC S9(9) BINARY VALUE 2005.                  1185
COPIED        10 MQRC-CHAR-ATTR-LENGTH-ERROR   PIC S9(9) BINARY VALUE 2006.                  1186
COPIED        10 MQRC-CHAR-ATTRS-ERROR         PIC S9(9) BINARY VALUE 2007.                  1187
COPIED        10 MQRC-CHAR-ATTRS-TOO-SHORT     PIC S9(9) BINARY VALUE 2008.                  1188
COPIED        10 MQRC-CONNECTION-BROKEN        PIC S9(9) BINARY VALUE 2009.                  1189
COPIED        10 MQRC-DATA-LENGTH-ERROR        PIC S9(9) BINARY VALUE 2010.                  1190
COPIED        10 MQRC-DYNAMIC-Q-NAME-ERROR     PIC S9(9) BINARY VALUE 2011.                  1191
COPIED        10 MQRC-ENVIRONMENT-ERROR        PIC S9(9) BINARY VALUE 2012.                  1192
COPIED        10 MQRC-EXPIRY-ERROR             PIC S9(9) BINARY VALUE 2013.                  1193
COPIED        10 MQRC-FEEDBACK-ERROR           PIC S9(9) BINARY VALUE 2014.                  1194
COPIED        10 MQRC-GET-INHIBITED            PIC S9(9) BINARY VALUE 2016.                  1195
COPIED        10 MQRC-HANDLE-NOT-AVAILABLE     PIC S9(9) BINARY VALUE 2017.                  1196
COPIED        10 MQRC-HCONN-ERROR              PIC S9(9) BINARY VALUE 2018.                  1197
COPIED        10 MQRC-HOBJ-ERROR               PIC S9(9) BINARY VALUE 2019.                  1198
COPIED        10 MQRC-INHIBIT-VALUE-ERROR      PIC S9(9) BINARY VALUE 2020.                  1199
COPIED        10 MQRC-INT-ATTR-COUNT-ERROR     PIC S9(9) BINARY VALUE 2021.                  1200
COPIED        10 MQRC-INT-ATTR-COUNT-TOO-SMALL PIC S9(9) BINARY VALUE 2022.                  1201
COPIED        10 MQRC-INT-ATTRS-ARRAY-ERROR    PIC S9(9) BINARY VALUE 2023.                  1202
COPIED        10 MQRC-SYNCPOINT-LIMIT-REACHED  PIC S9(9) BINARY VALUE 2024.                  1203
COPIED        10 MQRC-MAX-CONNS-LIMIT-REACHED  PIC S9(9) BINARY VALUE 2025.                  1204
COPIED        10 MQRC-MD-ERROR                 PIC S9(9) BINARY VALUE 2026.                  1205
COPIED        10 MQRC-MISSING-REPLY-TO-Q       PIC S9(9) BINARY VALUE 2027.                  1206
COPIED        10 MQRC-MSG-TYPE-ERROR           PIC S9(9) BINARY VALUE 2029.                  1207
COPIED        10 MQRC-MSG-TOO-BIG-FOR-Q        PIC S9(9) BINARY VALUE 2030.                  1208
COPIED        10 MQRC-MSG-TOO-BIG-FOR-Q-MGR    PIC S9(9) BINARY VALUE 2031.                  1209
COPIED        10 MQRC-NO-MSG-AVAILABLE         PIC S9(9) BINARY VALUE 2033.                  1210
```

Figure 23 MQPUT Sample Compiled Listing (Page 16 of 32)

```
COPIED        10 MQRC-NO-MSG-UNDER-CURSOR       PIC S9(9) BINARY VALUE 2034.                          1211
COPIED        10 MQRC-NOT-AUTHORIZED            PIC S9(9) BINARY VALUE 2035.                          1212
COPIED        10 MQRC-NOT-OPEN-FOR-BROWSE       PIC S9(9) BINARY VALUE 2036.                          1213
COPIED        10 MQRC-NOT-OPEN-FOR-INPUT        PIC S9(9) BINARY VALUE 2037.                          1214
COPIED        10 MQRC-NOT-OPEN-FOR-INQUIRE      PIC S9(9) BINARY VALUE 2038.                          1215
COPIED        10 MQRC-NOT-OPEN-FOR-OUTPUT       PIC S9(9) BINARY VALUE 2039.                          1216
COPIED        10 MQRC-NOT-OPEN-FOR-SET          PIC S9(9) BINARY VALUE 2040.                          1217
COPIED        10 MQRC-OBJECT-CHANGED            PIC S9(9) BINARY VALUE 2041.                          1218
COPIED        10 MQRC-OBJECT-IN-USE             PIC S9(9) BINARY VALUE 2042.                          1219
COPIED        10 MQRC-OBJECT-TYPE-ERROR         PIC S9(9) BINARY VALUE 2043.                          1220
COPIED        10 MQRC-OD-ERROR                  PIC S9(9) BINARY VALUE 2044.                          1221
COPIED        10 MQRC-OPTION-NOT-VALID-FOR-TYPE PIC S9(9) BINARY VALUE 2045.                          1222
COPIED        10 MQRC-OPTIONS-ERROR             PIC S9(9) BINARY VALUE 2046.                          1223

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    26
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQRC-PERSISTENCE-ERROR         PIC S9(9) BINARY VALUE 2047.                          1224
COPIED        10 MQRC-PERSISTENT-NOT-ALLOWED    PIC S9(9) BINARY VALUE 2048.                          1225
COPIED        10 MQRC-PRIORITY-EXCEEDS-MAXIMUM  PIC S9(9) BINARY VALUE 2049.                          1226
COPIED        10 MQRC-PRIORITY-ERROR            PIC S9(9) BINARY VALUE 2050.                          1227
COPIED        10 MQRC-PUT-INHIBITED             PIC S9(9) BINARY VALUE 2051.                          1228
COPIED        10 MQRC-Q-DELETED                 PIC S9(9) BINARY VALUE 2052.                          1229
COPIED        10 MQRC-Q-FULL                    PIC S9(9) BINARY VALUE 2053.                          1230
COPIED        10 MQRC-Q-NOT-EMPTY               PIC S9(9) BINARY VALUE 2055.                          1231
COPIED        10 MQRC-Q-SPACE-NOT-AVAILABLE     PIC S9(9) BINARY VALUE 2056.                          1232
COPIED        10 MQRC-Q-TYPE-ERROR              PIC S9(9) BINARY VALUE 2057.                          1233
COPIED        10 MQRC-Q-MGR-NAME-ERROR          PIC S9(9) BINARY VALUE 2058.                          1234
COPIED        10 MQRC-Q-MGR-NOT-AVAILABLE       PIC S9(9) BINARY VALUE 2059.                          1235
COPIED        10 MQRC-REPORT-OPTIONS-ERROR      PIC S9(9) BINARY VALUE 2061.                          1236
COPIED        10 MQRC-SECOND-MARK-NOT-ALLOWED   PIC S9(9) BINARY VALUE 2062.                          1237
COPIED        10 MQRC-SECURITY-ERROR            PIC S9(9) BINARY VALUE 2063.                          1238
COPIED        10 MQRC-SELECTOR-COUNT-ERROR      PIC S9(9) BINARY VALUE 2065.                          1239
COPIED        10 MQRC-SELECTOR-LIMIT-EXCEEDED   PIC S9(9) BINARY VALUE 2066.                          1240
COPIED        10 MQRC-SELECTOR-ERROR            PIC S9(9) BINARY VALUE 2067.                          1241
COPIED        10 MQRC-SELECTOR-NOT-FOR-TYPE     PIC S9(9) BINARY VALUE 2068.                          1242
COPIED        10 MQRC-SIGNAL-OUTSTANDING        PIC S9(9) BINARY VALUE 2069.                          1243
COPIED        10 MQRC-SIGNAL-REQUEST-ACCEPTED   PIC S9(9) BINARY VALUE 2070.                          1244
COPIED        10 MQRC-STORAGE-NOT-AVAILABLE     PIC S9(9) BINARY VALUE 2071.                          1245
COPIED        10 MQRC-SYNCPOINT-NOT-AVAILABLE   PIC S9(9) BINARY VALUE 2072.                          1246
COPIED        10 MQRC-TRIGGER-CONTROL-ERROR     PIC S9(9) BINARY VALUE 2075.                          1247
COPIED        10 MQRC-TRIGGER-DEPTH-ERROR       PIC S9(9) BINARY VALUE 2076.                          1248
COPIED        10 MQRC-TRIGGER-MSG-PRIORITY-ERR  PIC S9(9) BINARY VALUE 2077.                          1249
COPIED        10 MQRC-TRIGGER-TYPE-ERROR        PIC S9(9) BINARY VALUE 2078.                          1250
COPIED        10 MQRC-TRUNCATED-MSG-ACCEPTED    PIC S9(9) BINARY VALUE 2079.                          1251
COPIED        10 MQRC-TRUNCATED-MSG-FAILED      PIC S9(9) BINARY VALUE 2080.                          1252
COPIED        10 MQRC-UNKNOWN-ALIAS-BASE-Q      PIC S9(9) BINARY VALUE 2082.                          1253
COPIED        10 MQRC-UNKNOWN-OBJECT-NAME       PIC S9(9) BINARY VALUE 2085.                          1254
COPIED        10 MQRC-UNKNOWN-OBJECT-Q-MGR      PIC S9(9) BINARY VALUE 2086.                          1255
COPIED        10 MQRC-UNKNOWN-REMOTE-Q-MGR      PIC S9(9) BINARY VALUE 2087.                          1256
COPIED        10 MQRC-WAIT-INTERVAL-ERROR       PIC S9(9) BINARY VALUE 2090.                          1257
COPIED        10 MQRC-XMIT-Q-TYPE-ERROR         PIC S9(9) BINARY VALUE 2091.                          1258
COPIED        10 MQRC-XMIT-Q-USAGE-ERROR        PIC S9(9) BINARY VALUE 2092.                          1259
COPIED        10 MQRC-NOT-OPEN-FOR-PASS-ALL     PIC S9(9) BINARY VALUE 2093.                          1260
COPIED        10 MQRC-NOT-OPEN-FOR-PASS-IDENT   PIC S9(9) BINARY VALUE 2094.                          1261
COPIED        10 MQRC-NOT-OPEN-FOR-SET-ALL      PIC S9(9) BINARY VALUE 2095.                          1262
COPIED        10 MQRC-NOT-OPEN-FOR-SET-IDENT    PIC S9(9) BINARY VALUE 2096.                          1263
COPIED        10 MQRC-CONTEXT-HANDLE-ERROR      PIC S9(9) BINARY VALUE 2097.                          1264
COPIED        10 MQRC-CONTEXT-NOT-AVAILABLE     PIC S9(9) BINARY VALUE 2098.                          1265
COPIED        10 MQRC-SIGNAL1-ERROR             PIC S9(9) BINARY VALUE 2099.                          1266
COPIED        10 MQRC-OBJECT-ALREADY-EXISTS     PIC S9(9) BINARY VALUE 2100.                          1267
COPIED        10 MQRC-OBJECT-DAMAGED            PIC S9(9) BINARY VALUE 2101.                          1268
COPIED        10 MQRC-RESOURCE-PROBLEM          PIC S9(9) BINARY VALUE 2102.                          1269
COPIED        10 MQRC-ANOTHER-Q-MGR-CONNECTED   PIC S9(9) BINARY VALUE 2103.                          1270
COPIED        10 MQRC-UNKNOWN-REPORT-OPTION     PIC S9(9) BINARY VALUE 2104.                          1271
COPIED        10 MQRC-STORAGE-CLASS-ERROR       PIC S9(9) BINARY VALUE 2105.                          1272

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05           PAGE    27
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED        10 MQRC-COD-NOT-VALID-FOR-XCF-Q   PIC S9(9) BINARY VALUE 2106.                          1273
COPIED        10 MQRC-SUPPRESSED-BY-EXIT        PIC S9(9) BINARY VALUE 2109.                          1274
COPIED        10 MQRC-FORMAT-ERROR              PIC S9(9) BINARY VALUE 2110.                          1275
COPIED        10 MQRC-SOURCE-CCSID-ERROR        PIC S9(9) BINARY VALUE 2111.                          1276
COPIED        10 MQRC-SOURCE-INTEGER-ENC-ERROR  PIC S9(9) BINARY VALUE 2112.                          1277
COPIED        10 MQRC-SOURCE-DECIMAL-ENC-ERROR  PIC S9(9) BINARY VALUE 2113.                          1278
COPIED        10 MQRC-SOURCE-FLOAT-ENC-ERROR    PIC S9(9) BINARY VALUE 2114.                          1279
COPIED        10 MQRC-TARGET-CCSID-ERROR        PIC S9(9) BINARY VALUE 2115.                          1280
COPIED        10 MQRC-TARGET-INTEGER-ENC-ERROR  PIC S9(9) BINARY VALUE 2116.                          1281
COPIED        10 MQRC-TARGET-DECIMAL-ENC-ERROR  PIC S9(9) BINARY VALUE 2117.                          1282
COPIED        10 MQRC-TARGET-FLOAT-ENC-ERROR    PIC S9(9) BINARY VALUE 2118.                          1283
COPIED        10 MQRC-NOT-CONVERTED             PIC S9(9) BINARY VALUE 2119.                          1284
COPIED        10 MQRC-CONVERTED-MSG-TOO-BIG     PIC S9(9) BINARY VALUE 2120.                          1285
COPIED        10 MQRC-TRUNCATED                 PIC S9(9) BINARY VALUE 2120.                          1286
COPIED        10 MQRC-NO-EXTERNAL-PARTICIPANTS  PIC S9(9) BINARY VALUE 2121.                          1287
COPIED        10 MQRC-PARTICIPANT-NOT-AVAILABLE PIC S9(9) BINARY VALUE 2122.                          1288
COPIED        10 MQRC-OUTCOME-MIXED             PIC S9(9) BINARY VALUE 2123.                          1289
```

Figure 23 MQPUT Sample Compiled Listing (Page 17 of 32)

```
COPIED          10 MQRC-OUTCOME-PENDING           PIC S9(9) BINARY VALUE 2124.                    1290
COPIED          10 MQRC-BRIDGE-STARTED            PIC S9(9) BINARY VALUE 2125.                    1291
COPIED          10 MQRC-BRIDGE-STOPPED            PIC S9(9) BINARY VALUE 2126.                    1292
COPIED          10 MQRC-ADAPTER-STORAGE-SHORTAGE  PIC S9(9) BINARY VALUE 2127.                    1293
COPIED          10 MQRC-UOW-IN-PROGRESS           PIC S9(9) BINARY VALUE 2128.                    1294
COPIED          10 MQRC-ADAPTER-CONN-LOAD-ERROR   PIC S9(9) BINARY VALUE 2129.                    1295
COPIED          10 MQRC-ADAPTER-SERV-LOAD-ERROR   PIC S9(9) BINARY VALUE 2130.                    1296
COPIED          10 MQRC-ADAPTER-DEFS-ERROR        PIC S9(9) BINARY VALUE 2131.                    1297
COPIED          10 MQRC-ADAPTER-DEFS-LOAD-ERROR   PIC S9(9) BINARY VALUE 2132.                    1298
COPIED          10 MQRC-ADAPTER-CONV-LOAD-ERROR   PIC S9(9) BINARY VALUE 2133.                    1299
COPIED          10 MQRC-BO-ERROR                  PIC S9(9) BINARY VALUE 2134.                    1300
COPIED          10 MQRC-DH-ERROR                  PIC S9(9) BINARY VALUE 2135.                    1301
COPIED          10 MQRC-MULTIPLE-REASONS          PIC S9(9) BINARY VALUE 2136.                    1302
COPIED          10 MQRC-OPEN-FAILED               PIC S9(9) BINARY VALUE 2137.                    1303
COPIED          10 MQRC-ADAPTER-DISC-LOAD-ERROR   PIC S9(9) BINARY VALUE 2138.                    1304
COPIED          10 MQRC-CNO-ERROR                 PIC S9(9) BINARY VALUE 2139.                    1305
COPIED          10 MQRC-CICS-WAIT-FAILED          PIC S9(9) BINARY VALUE 2140.                    1306
COPIED          10 MQRC-DLH-ERROR                 PIC S9(9) BINARY VALUE 2141.                    1307
COPIED          10 MQRC-HEADER-ERROR              PIC S9(9) BINARY VALUE 2142.                    1308
COPIED          10 MQRC-SOURCE-LENGTH-ERROR       PIC S9(9) BINARY VALUE 2143.                    1309
COPIED          10 MQRC-TARGET-LENGTH-ERROR       PIC S9(9) BINARY VALUE 2144.                    1310
COPIED          10 MQRC-SOURCE-BUFFER-ERROR       PIC S9(9) BINARY VALUE 2145.                    1311
COPIED          10 MQRC-TARGET-BUFFER-ERROR       PIC S9(9) BINARY VALUE 2146.                    1312
COPIED          10 MQRC-IIH-ERROR                 PIC S9(9) BINARY VALUE 2148.                    1313
COPIED          10 MQRC-PCF-ERROR                 PIC S9(9) BINARY VALUE 2149.                    1314
COPIED          10 MQRC-DBCS-ERROR                PIC S9(9) BINARY VALUE 2150.                    1315
COPIED          10 MQRC-OBJECT-NAME-ERROR         PIC S9(9) BINARY VALUE 2152.                    1316
COPIED          10 MQRC-OBJECT-Q-MGR-NAME-ERROR   PIC S9(9) BINARY VALUE 2153.                    1317
COPIED          10 MQRC-RECS-PRESENT-ERROR        PIC S9(9) BINARY VALUE 2154.                    1318
COPIED          10 MQRC-OBJECT-RECORDS-ERROR      PIC S9(9) BINARY VALUE 2155.                    1319
COPIED          10 MQRC-RESPONSE-RECORDS-ERROR    PIC S9(9) BINARY VALUE 2156.                    1320
COPIED          10 MQRC-ASID-MISMATCH             PIC S9(9) BINARY VALUE 2157.                    1321

      COMPUTER ASSOCIATES       VISION:RESULTS  6.0 *             DATE  09/20/05            PAGE    28
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED          10 MQRC-PMO-RECORD-FLAGS-ERROR    PIC S9(9) BINARY VALUE 2158.                    1322
COPIED          10 MQRC-PUT-MSG-RECORDS-ERROR     PIC S9(9) BINARY VALUE 2159.                    1323
COPIED          10 MQRC-CONN-ID-IN-USE            PIC S9(9) BINARY VALUE 2160.                    1324
COPIED          10 MQRC-Q-MGR-QUIESCING           PIC S9(9) BINARY VALUE 2161.                    1325
COPIED          10 MQRC-Q-MGR-STOPPING            PIC S9(9) BINARY VALUE 2162.                    1326
COPIED          10 MQRC-DUPLICATE-RECOV-COORD     PIC S9(9) BINARY VALUE 2163.                    1327
COPIED          10 MQRC-PMO-ERROR                 PIC S9(9) BINARY VALUE 2173.                    1328
COPIED          10 MQRC-API-EXIT-NOT-FOUND        PIC S9(9) BINARY VALUE 2182.                    1329
COPIED          10 MQRC-API-EXIT-LOAD-ERROR       PIC S9(9) BINARY VALUE 2183.                    1330
COPIED          10 MQRC-REMOTE-Q-NAME-ERROR       PIC S9(9) BINARY VALUE 2184.                    1331
COPIED          10 MQRC-INCONSISTENT-PERSISTENCE  PIC S9(9) BINARY VALUE 2185.                    1332
COPIED          10 MQRC-GMO-ERROR                 PIC S9(9) BINARY VALUE 2186.                    1333
COPIED          10 MQRC-CICS-BRIDGE-RESTRICTION   PIC S9(9) BINARY VALUE 2187.                    1334
COPIED          10 MQRC-STOPPED-BY-CLUSTER-EXIT   PIC S9(9) BINARY VALUE 2188.                    1335
COPIED          10 MQRC-CLUSTER-RESOLUTION-ERROR  PIC S9(9) BINARY VALUE 2189.                    1336
COPIED          10 MQRC-CONVERTED-STRING-TOO-BIG  PIC S9(9) BINARY VALUE 2190.                    1337
COPIED          10 MQRC-TMC-ERROR                 PIC S9(9) BINARY VALUE 2191.                    1338
COPIED          10 MQRC-PAGESET-FULL              PIC S9(9) BINARY VALUE 2192.                    1339
COPIED          10 MQRC-STORAGE-MEDIUM-FULL       PIC S9(9) BINARY VALUE 2192.                    1340
COPIED          10 MQRC-PAGESET-ERROR             PIC S9(9) BINARY VALUE 2193.                    1341
COPIED          10 MQRC-NAME-NOT-VALID-FOR-TYPE   PIC S9(9) BINARY VALUE 2194.                    1342
COPIED          10 MQRC-UNEXPECTED-ERROR          PIC S9(9) BINARY VALUE 2195.                    1343
COPIED          10 MQRC-UNKNOWN-XMIT-Q            PIC S9(9) BINARY VALUE 2196.                    1344
COPIED          10 MQRC-UNKNOWN-DEF-XMIT-Q        PIC S9(9) BINARY VALUE 2197.                    1345
COPIED          10 MQRC-DEF-XMIT-Q-TYPE-ERROR     PIC S9(9) BINARY VALUE 2198.                    1346
COPIED          10 MQRC-DEF-XMIT-Q-USAGE-ERROR    PIC S9(9) BINARY VALUE 2199.                    1347
COPIED          10 MQRC-NAME-IN-USE               PIC S9(9) BINARY VALUE 2201.                    1348
COPIED          10 MQRC-CONNECTION-QUIESCING      PIC S9(9) BINARY VALUE 2202.                    1349
COPIED          10 MQRC-CONNECTION-STOPPING       PIC S9(9) BINARY VALUE 2203.                    1350
COPIED          10 MQRC-ADAPTER-NOT-AVAILABLE     PIC S9(9) BINARY VALUE 2204.                    1351
COPIED          10 MQRC-MSG-ID-ERROR              PIC S9(9) BINARY VALUE 2206.                    1352
COPIED          10 MQRC-CORREL-ID-ERROR           PIC S9(9) BINARY VALUE 2207.                    1353
COPIED          10 MQRC-FILE-SYSTEM-ERROR         PIC S9(9) BINARY VALUE 2208.                    1354
COPIED          10 MQRC-NO-MSG-LOCKED             PIC S9(9) BINARY VALUE 2209.                    1355
COPIED          10 MQRC-FILE-NOT-AUDITED          PIC S9(9) BINARY VALUE 2216.                    1356
COPIED          10 MQRC-CONNECTION-NOT-AUTHORIZED PIC S9(9) BINARY VALUE 2217.                    1357
COPIED          10 MQRC-MSG-TOO-BIG-FOR-CHANNEL   PIC S9(9) BINARY VALUE 2218.                    1358
COPIED          10 MQRC-CALL-IN-PROGRESS          PIC S9(9) BINARY VALUE 2219.                    1359
COPIED          10 MQRC-RMH-ERROR                 PIC S9(9) BINARY VALUE 2220.                    1360
COPIED          10 MQRC-Q-MGR-ACTIVE              PIC S9(9) BINARY VALUE 2222.                    1361
COPIED          10 MQRC-Q-MGR-NOT-ACTIVE          PIC S9(9) BINARY VALUE 2223.                    1362
COPIED          10 MQRC-Q-DEPTH-HIGH              PIC S9(9) BINARY VALUE 2224.                    1363
COPIED          10 MQRC-Q-DEPTH-LOW               PIC S9(9) BINARY VALUE 2225.                    1364
COPIED          10 MQRC-Q-SERVICE-INTERVAL-HIGH   PIC S9(9) BINARY VALUE 2226.                    1365
COPIED          10 MQRC-Q-SERVICE-INTERVAL-OK     PIC S9(9) BINARY VALUE 2227.                    1366
COPIED          10 MQRC-UNIT-OF-WORK-NOT-STARTED  PIC S9(9) BINARY VALUE 2232.                    1367
COPIED          10 MQRC-CHANNEL-AUTO-DEF-OK       PIC S9(9) BINARY VALUE 2233.                    1368
COPIED          10 MQRC-CHANNEL-AUTO-DEF-ERROR    PIC S9(9) BINARY VALUE 2234.                    1369
COPIED          10 MQRC-CFH-ERROR                 PIC S9(9) BINARY VALUE 2235.                    1370
```

Figure 23 MQPUT Sample Compiled Listing (Page 18 of 32)

```
       COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    29
       1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED         10 MQRC-CFIL-ERROR               PIC S9(9) BINARY VALUE 2236.                      1371
COPIED         10 MQRC-CFIN-ERROR               PIC S9(9) BINARY VALUE 2237.                      1372
COPIED         10 MQRC-CFSL-ERROR               PIC S9(9) BINARY VALUE 2238.                      1373
COPIED         10 MQRC-CFST-ERROR               PIC S9(9) BINARY VALUE 2239.                      1374
COPIED         10 MQRC-INCOMPLETE-GROUP         PIC S9(9) BINARY VALUE 2241.                      1375
COPIED         10 MQRC-INCOMPLETE-MSG           PIC S9(9) BINARY VALUE 2242.                      1376
COPIED         10 MQRC-INCONSISTENT-CCSIDS      PIC S9(9) BINARY VALUE 2243.                      1377
COPIED         10 MQRC-INCONSISTENT-ENCODINGS   PIC S9(9) BINARY VALUE 2244.                      1378
COPIED         10 MQRC-INCONSISTENT-UOW         PIC S9(9) BINARY VALUE 2245.                      1379
COPIED         10 MQRC-INVALID-MSG-UNDER-CURSOR PIC S9(9) BINARY VALUE 2246.                      1380
COPIED         10 MQRC-MATCH-OPTIONS-ERROR      PIC S9(9) BINARY VALUE 2247.                      1381
COPIED         10 MQRC-MDE-ERROR                PIC S9(9) BINARY VALUE 2248.                      1382
COPIED         10 MQRC-MSG-FLAGS-ERROR          PIC S9(9) BINARY VALUE 2249.                      1383
COPIED         10 MQRC-MSG-SEQ-NUMBER-ERROR     PIC S9(9) BINARY VALUE 2250.                      1384
COPIED         10 MQRC-OFFSET-ERROR             PIC S9(9) BINARY VALUE 2251.                      1385
COPIED         10 MQRC-ORIGINAL-LENGTH-ERROR    PIC S9(9) BINARY VALUE 2252.                      1386
COPIED         10 MQRC-SEGMENT-LENGTH-ZERO      PIC S9(9) BINARY VALUE 2253.                      1387
COPIED         10 MQRC-UOW-NOT-AVAILABLE        PIC S9(9) BINARY VALUE 2255.                      1388
COPIED         10 MQRC-WRONG-GMO-VERSION        PIC S9(9) BINARY VALUE 2256.                      1389
COPIED         10 MQRC-WRONG-MD-VERSION         PIC S9(9) BINARY VALUE 2257.                      1390
COPIED         10 MQRC-GROUP-ID-ERROR           PIC S9(9) BINARY VALUE 2258.                      1391
COPIED         10 MQRC-INCONSISTENT-BROWSE      PIC S9(9) BINARY VALUE 2259.                      1392
COPIED         10 MQRC-XQH-ERROR                PIC S9(9) BINARY VALUE 2260.                      1393
COPIED         10 MQRC-SRC-ENV-ERROR            PIC S9(9) BINARY VALUE 2261.                      1394
COPIED         10 MQRC-SRC-NAME-ERROR           PIC S9(9) BINARY VALUE 2262.                      1395
COPIED         10 MQRC-DEST-ENV-ERROR           PIC S9(9) BINARY VALUE 2263.                      1396
COPIED         10 MQRC-DEST-NAME-ERROR          PIC S9(9) BINARY VALUE 2264.                      1397
COPIED         10 MQRC-TM-ERROR                 PIC S9(9) BINARY VALUE 2265.                      1398
COPIED         10 MQRC-CLUSTER-EXIT-ERROR       PIC S9(9) BINARY VALUE 2266.                      1399
COPIED         10 MQRC-CLUSTER-EXIT-LOAD-ERROR  PIC S9(9) BINARY VALUE 2267.                      1400
COPIED         10 MQRC-CLUSTER-PUT-INHIBITED    PIC S9(9) BINARY VALUE 2268.                      1401
COPIED         10 MQRC-CLUSTER-RESOURCE-ERROR   PIC S9(9) BINARY VALUE 2269.                      1402
COPIED         10 MQRC-NO-DESTINATIONS-AVAILABLE PIC S9(9) BINARY VALUE 2270.                     1403
COPIED         10 MQRC-CONN-TAG-IN-USE          PIC S9(9) BINARY VALUE 2271.                      1404
COPIED         10 MQRC-PARTIALLY-CONVERTED      PIC S9(9) BINARY VALUE 2272.                      1405
COPIED         10 MQRC-CONNECTION-ERROR         PIC S9(9) BINARY VALUE 2273.                      1406
COPIED         10 MQRC-OPTION-ENVIRONMENT-ERROR PIC S9(9) BINARY VALUE 2274.                      1407
COPIED         10 MQRC-CD-ERROR                 PIC S9(9) BINARY VALUE 2277.                      1408
COPIED         10 MQRC-CLIENT-CONN-ERROR        PIC S9(9) BINARY VALUE 2278.                      1409
COPIED         10 MQRC-CHANNEL-STOPPED-BY-USER  PIC S9(9) BINARY VALUE 2279.                      1410
COPIED         10 MQRC-HCONFIG-ERROR            PIC S9(9) BINARY VALUE 2280.                      1411
COPIED         10 MQRC-FUNCTION-ERROR           PIC S9(9) BINARY VALUE 2281.                      1412
COPIED         10 MQRC-CHANNEL-STARTED          PIC S9(9) BINARY VALUE 2282.                      1413
COPIED         10 MQRC-CHANNEL-STOPPED          PIC S9(9) BINARY VALUE 2283.                      1414
COPIED         10 MQRC-CHANNEL-CONV-ERROR       PIC S9(9) BINARY VALUE 2284.                      1415
COPIED         10 MQRC-SERVICE-NOT-AVAILABLE    PIC S9(9) BINARY VALUE 2285.                      1416
COPIED         10 MQRC-INITIALIZATION-FAILED    PIC S9(9) BINARY VALUE 2286.                      1417
COPIED         10 MQRC-TERMINATION-FAILED       PIC S9(9) BINARY VALUE 2287.                      1418
COPIED         10 MQRC-UNKNOWN-Q-NAME           PIC S9(9) BINARY VALUE 2288.                      1419

       COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    30
       1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED         10 MQRC-SERVICE-ERROR            PIC S9(9) BINARY VALUE 2289.                      1420
COPIED         10 MQRC-Q-ALREADY-EXISTS         PIC S9(9) BINARY VALUE 2290.                      1421
COPIED         10 MQRC-USER-ID-NOT-AVAILABLE    PIC S9(9) BINARY VALUE 2291.                      1422
COPIED         10 MQRC-UNKNOWN-ENTITY           PIC S9(9) BINARY VALUE 2292.                      1423
COPIED         10 MQRC-UNKNOWN-AUTH-ENTITY      PIC S9(9) BINARY VALUE 2293.                      1424
COPIED         10 MQRC-UNKNOWN-REF-OBJECT       PIC S9(9) BINARY VALUE 2294.                      1425
COPIED         10 MQRC-CHANNEL-ACTIVATED        PIC S9(9) BINARY VALUE 2295.                      1426
COPIED         10 MQRC-CHANNEL-NOT-ACTIVATED    PIC S9(9) BINARY VALUE 2296.                      1427
COPIED         10 MQRC-UOW-CANCELED             PIC S9(9) BINARY VALUE 2297.                      1428
COPIED         10 MQRC-FUNCTION-NOT-SUPPORTED   PIC S9(9) BINARY VALUE 2298.                      1429
COPIED         10 MQRC-SELECTOR-TYPE-ERROR      PIC S9(9) BINARY VALUE 2299.                      1430
COPIED         10 MQRC-COMMAND-TYPE-ERROR       PIC S9(9) BINARY VALUE 2300.                      1431
COPIED         10 MQRC-MULTIPLE-INSTANCE-ERROR  PIC S9(9) BINARY VALUE 2301.                      1432
COPIED         10 MQRC-SYSTEM-ITEM-NOT-ALTERABLE PIC S9(9) BINARY VALUE 2302.                     1433
COPIED         10 MQRC-BAG-CONVERSION-ERROR     PIC S9(9) BINARY VALUE 2303.                      1434
COPIED         10 MQRC-SELECTOR-OUT-OF-RANGE    PIC S9(9) BINARY VALUE 2304.                      1435
COPIED         10 MQRC-SELECTOR-NOT-UNIQUE      PIC S9(9) BINARY VALUE 2305.                      1436
COPIED         10 MQRC-INDEX-NOT-PRESENT        PIC S9(9) BINARY VALUE 2306.                      1437
COPIED         10 MQRC-STRING-ERROR             PIC S9(9) BINARY VALUE 2307.                      1438
COPIED         10 MQRC-ENCODING-NOT-SUPPORTED   PIC S9(9) BINARY VALUE 2308.                      1439
COPIED         10 MQRC-SELECTOR-NOT-PRESENT     PIC S9(9) BINARY VALUE 2309.                      1440
COPIED         10 MQRC-OUT-SELECTOR-ERROR       PIC S9(9) BINARY VALUE 2310.                      1441
COPIED         10 MQRC-STRING-TRUNCATED         PIC S9(9) BINARY VALUE 2311.                      1442
COPIED         10 MQRC-SELECTOR-WRONG-TYPE      PIC S9(9) BINARY VALUE 2312.                      1443
COPIED         10 MQRC-INCONSISTENT-ITEM-TYPE   PIC S9(9) BINARY VALUE 2313.                      1444
COPIED         10 MQRC-INDEX-ERROR              PIC S9(9) BINARY VALUE 2314.                      1445
COPIED         10 MQRC-SYSTEM-BAG-NOT-ALTERABLE PIC S9(9) BINARY VALUE 2315.                      1446
COPIED         10 MQRC-ITEM-COUNT-ERROR         PIC S9(9) BINARY VALUE 2316.                      1447
COPIED         10 MQRC-FORMAT-NOT-SUPPORTED     PIC S9(9) BINARY VALUE 2317.                      1448
COPIED         10 MQRC-SELECTOR-NOT-SUPPORTED   PIC S9(9) BINARY VALUE 2318.                      1449
COPIED         10 MQRC-ITEM-VALUE-ERROR         PIC S9(9) BINARY VALUE 2319.                      1450
```

Figure 23 MQPUT Sample Compiled Listing (Page 19 of 32)

```
COPIED          10 MQRC-HBAG-ERROR                PIC S9(9) BINARY VALUE 2320.                          1451
COPIED          10 MQRC-PARAMETER-MISSING         PIC S9(9) BINARY VALUE 2321.                          1452
COPIED          10 MQRC-CMD-SERVER-NOT-AVAILABLE  PIC S9(9) BINARY VALUE 2322.                          1453
COPIED          10 MQRC-STRING-LENGTH-ERROR       PIC S9(9) BINARY VALUE 2323.                          1454
COPIED          10 MQRC-INQUIRY-COMMAND-ERROR     PIC S9(9) BINARY VALUE 2324.                          1455
COPIED          10 MQRC-NESTED-BAG-NOT-SUPPORTED  PIC S9(9) BINARY VALUE 2325.                          1456
COPIED          10 MQRC-BAG-WRONG-TYPE            PIC S9(9) BINARY VALUE 2326.                          1457
COPIED          10 MQRC-ITEM-TYPE-ERROR           PIC S9(9) BINARY VALUE 2327.                          1458
COPIED          10 MQRC-SYSTEM-BAG-NOT-DELETABLE  PIC S9(9) BINARY VALUE 2328.                          1459
COPIED          10 MQRC-SYSTEM-ITEM-NOT-DELETABLE PIC S9(9) BINARY VALUE 2329.                          1460
COPIED          10 MQRC-CODED-CHAR-SET-ID-ERROR   PIC S9(9) BINARY VALUE 2330.                          1461
COPIED          10 MQRC-MSG-TOKEN-ERROR           PIC S9(9) BINARY VALUE 2331.                          1462
COPIED          10 MQRC-MISSING-WIH               PIC S9(9) BINARY VALUE 2332.                          1463
COPIED          10 MQRC-WIH-ERROR                 PIC S9(9) BINARY VALUE 2333.                          1464
COPIED          10 MQRC-RFH-ERROR                 PIC S9(9) BINARY VALUE 2334.                          1465
COPIED          10 MQRC-RFH-STRING-ERROR          PIC S9(9) BINARY VALUE 2335.                          1466
COPIED          10 MQRC-RFH-COMMAND-ERROR         PIC S9(9) BINARY VALUE 2336.                          1467
COPIED          10 MQRC-RFH-PARM-ERROR            PIC S9(9) BINARY VALUE 2337.                          1468

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    31
       1--------------------VISION:RESULTS FREE FORM TEXT----------------------72          --------
COPIED          10 MQRC-RFH-DUPLICATE-PARM        PIC S9(9) BINARY VALUE 2338.                          1469
COPIED          10 MQRC-RFH-PARM-MISSING          PIC S9(9) BINARY VALUE 2339.                          1470
COPIED          10 MQRC-CHAR-CONVERSION-ERROR     PIC S9(9) BINARY VALUE 2340.                          1471
COPIED          10 MQRC-UCS2-CONVERSION-ERROR     PIC S9(9) BINARY VALUE 2341.                          1472
COPIED          10 MQRC-DB2-NOT-AVAILABLE         PIC S9(9) BINARY VALUE 2342.                          1473
COPIED          10 MQRC-OBJECT-NOT-UNIQUE         PIC S9(9) BINARY VALUE 2343.                          1474
COPIED          10 MQRC-CONN-TAG-NOT-RELEASED     PIC S9(9) BINARY VALUE 2344.                          1475
COPIED          10 MQRC-CF-NOT-AVAILABLE          PIC S9(9) BINARY VALUE 2345.                          1476
COPIED          10 MQRC-CF-STRUC-IN-USE           PIC S9(9) BINARY VALUE 2346.                          1477
COPIED          10 MQRC-CF-STRUC-LIST-HDR-IN-USE  PIC S9(9) BINARY VALUE 2347.                          1478
COPIED          10 MQRC-CF-STRUC-AUTH-FAILED      PIC S9(9) BINARY VALUE 2348.                          1479
COPIED          10 MQRC-CF-STRUC-ERROR            PIC S9(9) BINARY VALUE 2349.                          1480
COPIED          10 MQRC-CONN-TAG-NOT-USABLE       PIC S9(9) BINARY VALUE 2350.                          1481
COPIED          10 MQRC-GLOBAL-UOW-CONFLICT       PIC S9(9) BINARY VALUE 2351.                          1482
COPIED          10 MQRC-LOCAL-UOW-CONFLICT        PIC S9(9) BINARY VALUE 2352.                          1483
COPIED          10 MQRC-HANDLE-IN-USE-FOR-UOW     PIC S9(9) BINARY VALUE 2353.                          1484
COPIED          10 MQRC-UOW-ENLISTMENT-ERROR      PIC S9(9) BINARY VALUE 2354.                          1485
COPIED          10 MQRC-UOW-MIX-NOT-SUPPORTED     PIC S9(9) BINARY VALUE 2355.                          1486
COPIED          10 MQRC-WXP-ERROR                 PIC S9(9) BINARY VALUE 2356.                          1487
COPIED          10 MQRC-CURRENT-RECORD-ERROR      PIC S9(9) BINARY VALUE 2357.                          1488
COPIED          10 MQRC-NEXT-OFFSET-ERROR         PIC S9(9) BINARY VALUE 2358.                          1489
COPIED          10 MQRC-NO-RECORD-AVAILABLE       PIC S9(9) BINARY VALUE 2359.                          1490
COPIED          10 MQRC-OBJECT-LEVEL-INCOMPATIBLE PIC S9(9) BINARY VALUE 2360.                          1491
COPIED          10 MQRC-NEXT-RECORD-ERROR         PIC S9(9) BINARY VALUE 2361.                          1492
COPIED          10 MQRC-BACKOUT-THRESHOLD-REACHED PIC S9(9) BINARY VALUE 2362.                          1493
COPIED          10 MQRC-MSG-NOT-MATCHED           PIC S9(9) BINARY VALUE 2363.                          1494
COPIED          10 MQRC-JMS-FORMAT-ERROR          PIC S9(9) BINARY VALUE 2364.                          1495
COPIED          10 MQRC-SEGMENTS-NOT-SUPPORTED    PIC S9(9) BINARY VALUE 2365.                          1496
COPIED          10 MQRC-WRONG-CF-LEVEL            PIC S9(9) BINARY VALUE 2366.                          1497
COPIED          10 MQRC-CONFIG-CREATE-OBJECT      PIC S9(9) BINARY VALUE 2367.                          1498
COPIED          10 MQRC-CONFIG-CHANGE-OBJECT      PIC S9(9) BINARY VALUE 2368.                          1499
COPIED          10 MQRC-CONFIG-DELETE-OBJECT      PIC S9(9) BINARY VALUE 2369.                          1500
COPIED          10 MQRC-CONFIG-REFRESH-OBJECT     PIC S9(9) BINARY VALUE 2370.                          1501
COPIED          10 MQRC-CHANNEL-SSL-ERROR         PIC S9(9) BINARY VALUE 2371.                          1502
COPIED          10 MQRC-API-EXIT-ERROR            PIC S9(9) BINARY VALUE 2374.                          1503
COPIED          10 MQRC-API-EXIT-INIT-ERROR       PIC S9(9) BINARY VALUE 2375.                          1504
COPIED          10 MQRC-API-EXIT-TERM-ERROR       PIC S9(9) BINARY VALUE 2376.                          1505
COPIED          10 MQRC-EXIT-REASON-ERROR         PIC S9(9) BINARY VALUE 2377.                          1506
COPIED          10 MQRC-RESERVED-VALUE-ERROR      PIC S9(9) BINARY VALUE 2378.                          1507
COPIED          10 MQRC-NO-DATA-AVAILABLE         PIC S9(9) BINARY VALUE 2379.                          1508
COPIED          10 MQRC-SCO-ERROR                 PIC S9(9) BINARY VALUE 2380.                          1509
COPIED          10 MQRC-KEY-REPOSITORY-ERROR      PIC S9(9) BINARY VALUE 2381.                          1510
COPIED          10 MQRC-CRYPTO-HARDWARE-ERROR     PIC S9(9) BINARY VALUE 2382.                          1511
COPIED          10 MQRC-AUTH-INFO-REC-COUNT-ERROR PIC S9(9) BINARY VALUE 2383.                          1512
COPIED          10 MQRC-AUTH-INFO-REC-ERROR       PIC S9(9) BINARY VALUE 2384.                          1513
COPIED          10 MQRC-AIR-ERROR                 PIC S9(9) BINARY VALUE 2385.                          1514
COPIED          10 MQRC-AUTH-INFO-TYPE-ERROR      PIC S9(9) BINARY VALUE 2386.                          1515
COPIED          10 MQRC-AUTH-INFO-CONN-NAME-ERROR PIC S9(9) BINARY VALUE 2387.                          1516
COPIED          10 MQRC-LDAP-USER-NAME-ERROR      PIC S9(9) BINARY VALUE 2388.                          1517

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    32
       1--------------------VISION:RESULTS FREE FORM TEXT----------------------72          --------
COPIED          10 MQRC-LDAP-USER-NAME-LENGTH-ERR PIC S9(9) BINARY VALUE 2389.                          1518
COPIED          10 MQRC-LDAP-PASSWORD-ERROR       PIC S9(9) BINARY VALUE 2390.                          1519
COPIED          10 MQRC-SSL-ALREADY-INITIALIZED   PIC S9(9) BINARY VALUE 2391.                          1520
COPIED          10 MQRC-SSL-CONFIG-ERROR          PIC S9(9) BINARY VALUE 2392.                          1521
COPIED          10 MQRC-SSL-INITIALIZATION-ERROR  PIC S9(9) BINARY VALUE 2393.                          1522
COPIED          10 MQRC-Q-INDEX-TYPE-ERROR        PIC S9(9) BINARY VALUE 2394.                          1523
COPIED          10 MQRC-SSL-NOT-ALLOWED           PIC S9(9) BINARY VALUE 2396.                          1524
COPIED          10 MQRC-JSSE-ERROR                PIC S9(9) BINARY VALUE 2397.                          1525
COPIED          10 MQRC-SSL-PEER-NAME-MISMATCH    PIC S9(9) BINARY VALUE 2398.                          1526
COPIED          10 MQRC-SSL-PEER-NAME-ERROR       PIC S9(9) BINARY VALUE 2399.                          1527
COPIED          10 MQRC-UNSUPPORTED-CIPHER-SUITE  PIC S9(9) BINARY VALUE 2400.                          1528
COPIED          10 MQRC-SSL-CERTIFICATE-REVOKED   PIC S9(9) BINARY VALUE 2401.                          1529
```

Figure 23 MQPUT Sample Compiled Listing (Page 20 of 32)

```
COPIED        10 MQRC-SSL-CERT-STORE-ERROR      PIC S9(9) BINARY VALUE 2402.                    1530
COPIED        10 MQRC-REOPEN-EXCL-INPUT-ERROR   PIC S9(9) BINARY VALUE 6100.                    1531
COPIED        10 MQRC-REOPEN-INQUIRE-ERROR      PIC S9(9) BINARY VALUE 6101.                    1532
COPIED        10 MQRC-REOPEN-SAVED-CONTEXT-ERR  PIC S9(9) BINARY VALUE 6102.                    1533
COPIED        10 MQRC-REOPEN-TEMPORARY-Q-ERROR  PIC S9(9) BINARY VALUE 6103.                    1534
COPIED        10 MQRC-ATTRIBUTE-LOCKED          PIC S9(9) BINARY VALUE 6104.                    1535
COPIED        10 MQRC-CURSOR-NOT-VALID          PIC S9(9) BINARY VALUE 6105.                    1536
COPIED        10 MQRC-ENCODING-ERROR            PIC S9(9) BINARY VALUE 6106.                    1537
COPIED        10 MQRC-STRUC-ID-ERROR            PIC S9(9) BINARY VALUE 6107.                    1538
COPIED        10 MQRC-NULL-POINTER              PIC S9(9) BINARY VALUE 6108.                    1539
COPIED        10 MQRC-NO-CONNECTION-REFERENCE   PIC S9(9) BINARY VALUE 6109.                    1540
COPIED        10 MQRC-NO-BUFFER                 PIC S9(9) BINARY VALUE 6110.                    1541
COPIED        10 MQRC-BINARY-DATA-LENGTH-ERROR  PIC S9(9) BINARY VALUE 6111.                    1542
COPIED        10 MQRC-BUFFER-NOT-AUTOMATIC      PIC S9(9) BINARY VALUE 6112.                    1543
COPIED        10 MQRC-INSUFFICIENT-BUFFER       PIC S9(9) BINARY VALUE 6113.                    1544
COPIED        10 MQRC-INSUFFICIENT-DATA         PIC S9(9) BINARY VALUE 6114.                    1545
COPIED        10 MQRC-DATA-TRUNCATED            PIC S9(9) BINARY VALUE 6115.                    1546
COPIED        10 MQRC-ZERO-LENGTH               PIC S9(9) BINARY VALUE 6116.                    1547
COPIED        10 MQRC-NEGATIVE-LENGTH           PIC S9(9) BINARY VALUE 6117.                    1548
COPIED        10 MQRC-NEGATIVE-OFFSET           PIC S9(9) BINARY VALUE 6118.                    1549
COPIED        10 MQRC-INCONSISTENT-FORMAT       PIC S9(9) BINARY VALUE 6119.                    1550
COPIED        10 MQRC-INCONSISTENT-OBJECT-STATE PIC S9(9) BINARY VALUE 6120.                    1551
COPIED        10 MQRC-CONTEXT-OBJECT-NOT-VALID  PIC S9(9) BINARY VALUE 6121.                    1552
COPIED        10 MQRC-CONTEXT-OPEN-ERROR        PIC S9(9) BINARY VALUE 6122.                    1553
COPIED        10 MQRC-STRUC-LENGTH-ERROR        PIC S9(9) BINARY VALUE 6123.                    1554
COPIED        10 MQRC-NOT-CONNECTED             PIC S9(9) BINARY VALUE 6124.                    1555
COPIED        10 MQRC-NOT-OPEN                  PIC S9(9) BINARY VALUE 6125.                    1556
COPIED        10 MQRC-DISTRIBUTION-LIST-EMPTY   PIC S9(9) BINARY VALUE 6126.                    1557
COPIED        10 MQRC-INCONSISTENT-OPEN-OPTIONS PIC S9(9) BINARY VALUE 6127.                    1558
COPIED        10 MQRC-WRONG-VERSION             PIC S9(9) BINARY VALUE 6128.                    1559
COPIED        10 MQRC-REFERENCE-ERROR           PIC S9(9) BINARY VALUE 6129.                    1560
COPIED                                                                                          1561
COPIED                                                                                          1562
COPIED      ****************************************************************                  1563
COPIED      **  Values Related to Queue Attributes                        **                  1564
COPIED      ****************************************************************                  1565
COPIED                                                                                          1566

       COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    33
       1-------------------VISION:RESULTS FREE FORM TEXT---------------------72        --------
COPIED      **   Queue Types                                                                    1567
COPIED        10 MQQT-LOCAL   PIC S9(9) BINARY VALUE 1.                                       1568
COPIED        10 MQQT-MODEL   PIC S9(9) BINARY VALUE 2.                                       1569
COPIED        10 MQQT-ALIAS   PIC S9(9) BINARY VALUE 3.                                       1570
COPIED        10 MQQT-REMOTE  PIC S9(9) BINARY VALUE 6.                                       1571
COPIED        10 MQQT-CLUSTER PIC S9(9) BINARY VALUE 7.                                       1572
COPIED                                                                                          1573
COPIED      **   Cluster Queue Types                                                            1574
COPIED        10 MQCQT-LOCAL-Q     PIC S9(9) BINARY VALUE 1.                                  1575
COPIED        10 MQCQT-ALIAS-Q     PIC S9(9) BINARY VALUE 2.                                  1576
COPIED        10 MQCQT-REMOTE-Q    PIC S9(9) BINARY VALUE 3.                                  1577
COPIED        10 MQCQT-Q-MGR-ALIAS PIC S9(9) BINARY VALUE 4.                                  1578
COPIED                                                                                          1579
COPIED      **   Extended Queue Types                                                           1580
COPIED        10 MQQT-ALL PIC S9(9) BINARY VALUE 1001.                                        1581
COPIED                                                                                          1582
COPIED      **   Queue Definition Types                                                         1583
COPIED        10 MQQDT-PREDEFINED        PIC S9(9) BINARY VALUE 1.                          1584
COPIED        10 MQQDT-PERMANENT-DYNAMIC PIC S9(9) BINARY VALUE 2.                          1585
COPIED        10 MQQDT-TEMPORARY-DYNAMIC PIC S9(9) BINARY VALUE 3.                          1586
COPIED        10 MQQDT-SHARED-DYNAMIC    PIC S9(9) BINARY VALUE 4.                          1587
COPIED                                                                                          1588
COPIED      **   Inhibit Get                                                                    1589
COPIED        10 MQQA-GET-INHIBITED PIC S9(9) BINARY VALUE 1.                                 1590
COPIED        10 MQQA-GET-ALLOWED   PIC S9(9) BINARY VALUE 0.                                 1591
COPIED                                                                                          1592
COPIED      **   Inhibit Put                                                                    1593
COPIED        10 MQQA-PUT-INHIBITED PIC S9(9) BINARY VALUE 1.                                 1594
COPIED        10 MQQA-PUT-ALLOWED   PIC S9(9) BINARY VALUE 0.                                 1595
COPIED                                                                                          1596
COPIED      **   Queue Shareability                                                             1597
COPIED        10 MQQA-SHAREABLE     PIC S9(9) BINARY VALUE 1.                                 1598
COPIED        10 MQQA-NOT-SHAREABLE PIC S9(9) BINARY VALUE 0.                                 1599
COPIED                                                                                          1600
COPIED      **   Back-Out Hardening                                                             1601
COPIED        10 MQQA-BACKOUT-HARDENED     PIC S9(9) BINARY VALUE 1.                        1602
COPIED        10 MQQA-BACKOUT-NOT-HARDENED PIC S9(9) BINARY VALUE 0.                        1603
COPIED                                                                                          1604
COPIED      **   Message Delivery Sequence                                                      1605
COPIED        10 MQMDS-PRIORITY PIC S9(9) BINARY VALUE 0.                                     1606
COPIED        10 MQMDS-FIFO     PIC S9(9) BINARY VALUE 1.                                     1607
COPIED                                                                                          1608
COPIED      **   Trigger Control                                                                1609
COPIED        10 MQTC-OFF PIC S9(9) BINARY VALUE 0.                                           1610
COPIED        10 MQTC-ON  PIC S9(9) BINARY VALUE 1.                                           1611
```

Figure 23 MQPUT Sample Compiled Listing (Page 21 of 32)

```
COPIED                                                                          1612
COPIED      **   Trigger Types                                                  1613
COPIED        10 MQTT-NONE  PIC S9(9) BINARY VALUE 0.                          1614
COPIED        10 MQTT-FIRST PIC S9(9) BINARY VALUE 1.                          1615

       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    34
       1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED        10 MQTT-EVERY PIC S9(9) BINARY VALUE 2.                          1616
COPIED        10 MQTT-DEPTH PIC S9(9) BINARY VALUE 3.                          1617
COPIED                                                                          1618
COPIED      **   Queue Usage                                                    1619
COPIED        10 MQUS-NORMAL       PIC S9(9) BINARY VALUE 0.                  1620
COPIED        10 MQUS-TRANSMISSION PIC S9(9) BINARY VALUE 1.                  1621
COPIED                                                                          1622
COPIED      **   Distribution Lists                                             1623
COPIED        10 MQDL-SUPPORTED     PIC S9(9) BINARY VALUE 1.                 1624
COPIED        10 MQDL-NOT-SUPPORTED PIC S9(9) BINARY VALUE 0.                 1625
COPIED                                                                          1626
COPIED      **   Index Type                                                     1627
COPIED        10 MQIT-NONE      PIC S9(9) BINARY VALUE 0.                     1628
COPIED        10 MQIT-MSG-ID    PIC S9(9) BINARY VALUE 1.                     1629
COPIED        10 MQIT-CORREL-ID PIC S9(9) BINARY VALUE 2.                     1630
COPIED        10 MQIT-MSG-TOKEN PIC S9(9) BINARY VALUE 4.                     1631
COPIED        10 MQIT-GROUP-ID  PIC S9(9) BINARY VALUE 5.                     1632
COPIED                                                                          1633
COPIED      **   Default Bind                                                   1634
COPIED        10 MQBND-BIND-ON-OPEN   PIC S9(9) BINARY VALUE 0.               1635
COPIED        10 MQBND-BIND-NOT-FIXED PIC S9(9) BINARY VALUE 1.               1636
COPIED                                                                          1637
COPIED      **   Queue Sharing Group Disposition                                1638
COPIED        10 MQQSGD-Q-MGR  PIC S9(9) BINARY VALUE 0.                      1639
COPIED        10 MQQSGD-COPY   PIC S9(9) BINARY VALUE 1.                      1640
COPIED        10 MQQSGD-SHARED PIC S9(9) BINARY VALUE 2.                      1641
COPIED        10 MQQSGD-GROUP  PIC S9(9) BINARY VALUE 3.                      1642
COPIED                                                                          1643
COPIED                                                                          1644
COPIED      ****************************************************************   1645
COPIED      **  Values Related to Namelist Attributes                     **   1646
COPIED      ****************************************************************   1647
COPIED                                                                          1648
COPIED      **   Name Count                                                     1649
COPIED        10 MQNC-MAX-NAMELIST-NAME-COUNT PIC S9(9) BINARY VALUE 256.     1650
COPIED                                                                          1651
COPIED      **   Namelist Type                                                  1652
COPIED        10 MQNT-NONE      PIC S9(9) BINARY VALUE 0.                     1653
COPIED        10 MQNT-Q         PIC S9(9) BINARY VALUE 1.                     1654
COPIED        10 MQNT-CLUSTER   PIC S9(9) BINARY VALUE 2.                     1655
COPIED        10 MQNT-AUTH-INFO PIC S9(9) BINARY VALUE 4.                     1656
COPIED        10 MQNT-ALL       PIC S9(9) BINARY VALUE 1001.                  1657
COPIED                                                                          1658
COPIED                                                                          1659
COPIED      ****************************************************************   1660
COPIED      **  Values Related to Process-Definition Attributes           **   1661
COPIED      ****************************************************************   1662
COPIED                                                                          1663
COPIED      **   Application Type                                               1664
       COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    35
       1--------------------VISION:RESULTS FREE FORM TEXT---------------------72         --------
COPIED      **   See values for "Put Application Type" under MQMD               1665
COPIED                                                                          1666
COPIED                                                                          1667
COPIED      ****************************************************************   1668
COPIED      **  Values Related to Authentication-Information Attributes   **   1669
COPIED      ****************************************************************   1670
COPIED                                                                          1671
COPIED      **   Authentication Information Type                                1672
COPIED      **   See values for "Authentication Information Type" under MQAIR  1673
COPIED                                                                          1674
COPIED                                                                          1675
COPIED      ****************************************************************   1676
COPIED      **  Values Related to Queue-Manager Attributes                **   1677
COPIED      ****************************************************************   1678
COPIED                                                                          1679
COPIED      **   Channel Auto Definition                                        1680
COPIED        10 MQCHAD-DISABLED PIC S9(9) BINARY VALUE 0.                     1681
COPIED        10 MQCHAD-ENABLED  PIC S9(9) BINARY VALUE 1.                     1682
COPIED                                                                          1683
COPIED      **   Command Level                                                  1684
COPIED        10 MQCMDL-LEVEL-1   PIC S9(9) BINARY VALUE 100.                  1685
COPIED        10 MQCMDL-LEVEL-101 PIC S9(9) BINARY VALUE 101.                  1686
COPIED        10 MQCMDL-LEVEL-110 PIC S9(9) BINARY VALUE 110.                  1687
COPIED        10 MQCMDL-LEVEL-114 PIC S9(9) BINARY VALUE 114.                  1688
COPIED        10 MQCMDL-LEVEL-120 PIC S9(9) BINARY VALUE 120.                  1689
COPIED        10 MQCMDL-LEVEL-200 PIC S9(9) BINARY VALUE 200.                  1690
COPIED        10 MQCMDL-LEVEL-201 PIC S9(9) BINARY VALUE 201.                  1691
```

Figure 23 MQPUT Sample Compiled Listing (Page 22 of 32)

```
COPIED        10 MQCMDL-LEVEL-210 PIC S9(9) BINARY VALUE 210.                    1692
COPIED        10 MQCMDL-LEVEL-220 PIC S9(9) BINARY VALUE 220.                    1693
COPIED        10 MQCMDL-LEVEL-221 PIC S9(9) BINARY VALUE 221.                    1694
COPIED        10 MQCMDL-LEVEL-320 PIC S9(9) BINARY VALUE 320.                    1695
COPIED        10 MQCMDL-LEVEL-420 PIC S9(9) BINARY VALUE 420.                    1696
COPIED        10 MQCMDL-LEVEL-500 PIC S9(9) BINARY VALUE 500.                    1697
COPIED        10 MQCMDL-LEVEL-510 PIC S9(9) BINARY VALUE 510.                    1698
COPIED        10 MQCMDL-LEVEL-520 PIC S9(9) BINARY VALUE 520.                    1699
COPIED        10 MQCMDL-LEVEL-530 PIC S9(9) BINARY VALUE 530.                    1700
COPIED                                                                           1701
COPIED      **   Distribution Lists                                              1702
COPIED      **   See values for "Distribution Lists" under Queue Attributes      1703
COPIED                                                                           1704
COPIED      **   Expiration Scan Interval                                        1705
COPIED        10 MQEXPI-OFF PIC S9(9) BINARY VALUE 0.                            1706
COPIED                                                                           1707
COPIED      **   Intra-Group Queuing                                             1708
COPIED        10 MQIGQ-DISABLED PIC S9(9) BINARY VALUE 0.                        1709
COPIED        10 MQIGQ-ENABLED  PIC S9(9) BINARY VALUE 1.                        1710
COPIED                                                                           1711
COPIED      **   Intra-Group Queuing Put Authority                               1712
COPIED        10 MQIGQPA-DEFAULT         PIC S9(9) BINARY VALUE 1.             1713

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    36
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
COPIED        10 MQIGQPA-CONTEXT         PIC S9(9) BINARY VALUE 2.             1714
COPIED        10 MQIGQPA-ONLY-IGQ        PIC S9(9) BINARY VALUE 3.             1715
COPIED        10 MQIGQPA-ALTERNATE-OR-IGQ PIC S9(9) BINARY VALUE 4.            1716
COPIED                                                                           1717
COPIED      **   Platform                                                        1718
COPIED        10 MQPL-MVS        PIC S9(9) BINARY VALUE 1.                     1719
COPIED        10 MQPL-OS390      PIC S9(9) BINARY VALUE 1.                     1720
COPIED        10 MQPL-ZOS        PIC S9(9) BINARY VALUE 1.                     1721
COPIED        10 MQPL-OS2        PIC S9(9) BINARY VALUE 2.                     1722
COPIED        10 MQPL-AIX        PIC S9(9) BINARY VALUE 3.                     1723
COPIED        10 MQPL-UNIX       PIC S9(9) BINARY VALUE 3.                     1724
COPIED        10 MQPL-OS400      PIC S9(9) BINARY VALUE 4.                     1725
COPIED        10 MQPL-WINDOWS    PIC S9(9) BINARY VALUE 5.                     1726
COPIED        10 MQPL-WINDOWS-NT PIC S9(9) BINARY VALUE 11.                    1727
COPIED        10 MQPL-VMS        PIC S9(9) BINARY VALUE 12.                    1728
COPIED        10 MQPL-NSK        PIC S9(9) BINARY VALUE 13.                    1729
COPIED        10 MQPL-NATIVE     PIC S9(9) BINARY VALUE 1.                     1730
COPIED                                                                           1731
COPIED      **   Syncpoint Availability                                          1732
COPIED        10 MQSP-AVAILABLE     PIC S9(9) BINARY VALUE 1.                  1733
COPIED        10 MQSP-NOT-AVAILABLE PIC S9(9) BINARY VALUE 0.                  1734
WARNING   DYL-187W    ZERO LENGTH DATANAME OF  CHARATIRS  MUST NOT BE REFERENCED
COPIED                                                                           1735
COPIED                                                                           1736
COPIED      ****************************************************************** 1737
COPIED      **  End of CMQV                                                   ** 1738
COPIED      ****************************************************************** 1739
      WORKAREA                                                                   1740
      COPY CMQCNOV COBOL                                                         1741
COPIED      ****************************************************************** 1742
COPIED      **                                                              ** 1743
COPIED      **                WebSphere MQ for z/OS                         ** 1744
COPIED      **                                                              ** 1745
COPIED      **  FILE NAME:      CMQCNOV                                     ** 1746
COPIED      **                                                              ** 1747
COPIED      **  DESCRIPTION:    Connect Options Structure                   ** 1748
COPIED      **                                                              ** 1749
COPIED      ****************************************************************** 1750
COPIED      **  @START_COPYRIGHT@                                           ** 1751
COPIED      **  Statement:     Licensed Materials - Property of IBM         ** 1752
COPIED      **                                                              ** 1753
COPIED      **                5655-F10                                      ** 1754
COPIED      **                (C) Copyright IBM Corporation. 1997, 2002     ** 1755
COPIED      **                                                              ** 1756
COPIED      **  Status:        Version 5 Release 3                         ** 1757
COPIED      **  @END_COPYRIGHT@                                             ** 1758
COPIED      ****************************************************************** 1759
COPIED      **                                                              ** 1760
COPIED      **  FUNCTION:       This file declares the structure MQCNO,     ** 1761

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE    37
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72        --------
COPIED      **                which is used by the main MQI.                ** 1762
COPIED      **                                                              ** 1763
COPIED      **  PROCESSOR:      COBOL                                       ** 1764
COPIED      **                                                              ** 1765
COPIED      ****************************************************************** 1766
COPIED                                                                           1767
COPIED      **   MQCNO structure                                                 1768
COPIED        10 MQCNO.                                                        1769
```

Figure 23 MQPUT Sample Compiled Listing (Page 23 of 32)

```
COPIED      **   Structure identifier                                          1770
COPIED        15 MQCNO-STRUCID          PIC X(4) VALUE 'CNO '.               1771
COPIED      **   Structure version number                                      1772
COPIED        15 MQCNO-VERSION          PIC S9(9) BINARY VALUE 1.            1773
COPIED      **   Options that control the action of MQCONNX                    1774
COPIED        15 MQCNO-OPTIONS          PIC S9(9) BINARY VALUE 0.            1775
COPIED      **   Offset of MQCD structure for client connection               1776
COPIED        15 MQCNO-CLIENTCONNOFFSET PIC S9(9) BINARY VALUE 0.            1777
COPIED      **   Address of MQCD structure for client connection              1778
COPIED        15 MQCNO-CLIENTCONNPTR    POINTER VALUE NULL.                  1779
COPIED      **   Queue-manager connection tag                                 1780
COPIED        15 MQCNO-CONNTAG          PIC X(128) VALUE LOW-VALUES.         1781
COPIED                                                                        1782
COPIED      ******************************************************************1783
COPIED      ** End of CMQCNOV                                              **1784
COPIED      ******************************************************************1785
      WORKAREA                                                                1786
      COPY CMQMDV COBOL                                                       1787
COPIED      ******************************************************************1788
COPIED      **                                                              **1789
COPIED      **               WebSphere MQ for z/OS                          **1790
COPIED      **                                                              **1791
COPIED      ** FILE NAME:     CMQMDV                                        **1792
COPIED      **                                                              **1793
COPIED      ** DESCRIPTION:   Message Descriptor Structure                  **1794
COPIED      **                                                              **1795
COPIED      ******************************************************************1796
COPIED      ** @START_COPYRIGHT@                                            **1797
COPIED      ** Statement:     Licensed Materials - Property of IBM          **1798
COPIED      **                                                              **1799
COPIED      **               5655-F10                                       **1800
COPIED      **               (C) Copyright IBM Corporation. 1993, 2002      **1801
COPIED      **                                                              **1802
COPIED      ** Status:        Version 5 Release 3                           **1803
COPIED      ** @END_COPYRIGHT@                                              **1804
COPIED      ******************************************************************1805
COPIED      **                                                              **1806
COPIED      ** FUNCTION:      This file declares the structure MQMD,        **1807
COPIED      **               which is used by the main MQI.                 **1808
COPIED      **                                                              **1809
COPIED      ** PROCESSOR:     COBOL                                         **1810

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    38
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED      **                                                              **1811
COPIED      ******************************************************************1812
COPIED                                                                        1813
COPIED      **  MQMD structure                                              1814
COPIED        10 MQMD.                                                      1815
COPIED      **   Structure identifier                                       1816
COPIED        15 MQMD-STRUCID         PIC X(4) VALUE 'MD  '.               1817
COPIED      **   Structure version number                                   1818
COPIED        15 MQMD-VERSION         PIC S9(9) BINARY VALUE 1.            1819
COPIED      **   Options for report messages                                1820
COPIED        15 MQMD-REPORT          PIC S9(9) BINARY VALUE 0.            1821
COPIED      **   Message type                                               1822
COPIED        15 MQMD-MSGTYPE         PIC S9(9) BINARY VALUE 8.            1823
COPIED      **   Message lifetime                                           1824
COPIED        15 MQMD-EXPIRY          PIC S9(9) BINARY VALUE -1.           1825
COPIED      **   Feedback or reason code                                    1826
COPIED        15 MQMD-FEEDBACK        PIC S9(9) BINARY VALUE 0.            1827
COPIED      **   Numeric encoding of message data                           1828
COPIED        15 MQMD-ENCODING        PIC S9(9) BINARY VALUE 785.          1829
COPIED      **   Character set identifier of message data                   1830
COPIED        15 MQMD-CODEDCHARSETID  PIC S9(9) BINARY VALUE 0.            1831
COPIED      **   Format name of message data                                1832
COPIED        15 MQMD-FORMAT          PIC X(8) VALUE SPACES.               1833
COPIED      **   Message priority                                           1834
COPIED        15 MQMD-PRIORITY        PIC S9(9) BINARY VALUE -1.           1835
COPIED      **   Message persistence                                        1836
COPIED        15 MQMD-PERSISTENCE     PIC S9(9) BINARY VALUE 2.            1837
COPIED      **   Message identifier                                         1838
COPIED        15 MQMD-MSGID           PIC X(24) VALUE LOW-VALUES.          1839
COPIED      **   Correlation identifier                                     1840
COPIED        15 MQMD-CORRELID        PIC X(24) VALUE LOW-VALUES.          1841
COPIED      **   Backout counter                                            1842
COPIED        15 MQMD-BACKOUTCOUNT    PIC S9(9) BINARY VALUE 0.            1843
COPIED      **   Name of reply queue                                        1844
COPIED        15 MQMD-REPLYTOQ        PIC X(48) VALUE SPACES.              1845
COPIED      **   Name of reply queue manager                                1846
COPIED        15 MQMD-REPLYTOQMGR     PIC X(48) VALUE SPACES.              1847
COPIED      **   User identifier                                            1848
COPIED        15 MQMD-USERIDENTIFIER  PIC X(12) VALUE SPACES.              1849
COPIED      **   Accounting token                                           1850
COPIED        15 MQMD-ACCOUNTINGTOKEN PIC X(32) VALUE LOW-VALUES.          1851
```

Figure 23 MQPUT Sample Compiled Listing (Page 24 of 32)

```
COPIED      **    Application data relating to identity                          1852
COPIED        15 MQMD-APPLIDENTITYDATA PIC X(32) VALUE SPACES.                   1853
COPIED      **    Type of application that put the message                       1854
COPIED        15 MQMD-PUTAPPLTYPE      PIC S9(9) BINARY VALUE 0.                 1855
COPIED      **    Name of application that put the message                       1856
COPIED        15 MQMD-PUTAPPLNAME      PIC X(28) VALUE SPACES.                   1857
COPIED      **    Date when message was put                                      1858
COPIED        15 MQMD-PUTDATE          PIC X(8) VALUE SPACES.                    1859

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    39
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED      **    Time when message was put                                      1860
COPIED        15 MQMD-PUTTIME          PIC X(8) VALUE SPACES.                    1861
COPIED      **    Application data relating to origin                            1862
COPIED        15 MQMD-APPLORIGINDATA   PIC X(4) VALUE SPACES.                    1863
COPIED      **    Group identifier                                               1864
COPIED        15 MQMD-GROUPID          PIC X(24) VALUE LOW-VALUES.               1865
COPIED      **    Sequence number of logical message within group                1866
COPIED        15 MQMD-MSGSEQNUMBER     PIC S9(9) BINARY VALUE 1.                 1867
COPIED      **    Offset of data in physical message from start of logical       1868
COPIED      **    message                                                        1869
COPIED        15 MQMD-OFFSET           PIC S9(9) BINARY VALUE 0.                 1870
COPIED      **    Message flags                                                  1871
COPIED        15 MQMD-MSGFLAGS         PIC S9(9) BINARY VALUE 0.                 1872
COPIED      **    Length of original message                                     1873
COPIED        15 MQMD-ORIGINALLENGTH   PIC S9(9) BINARY VALUE -1.                1874
COPIED                                                                           1875
COPIED      ******************************************************************** 1876
COPIED      **  End of CMQMDV                                               **   1877
COPIED      ******************************************************************** 1878
      WORKAREA                                                                   1879
      COPY CMQGMOV COBOL                                                         1880
COPIED      ******************************************************************** 1881
COPIED      **                                                              **   1882
COPIED      **                WebSphere MQ for z/OS                         **   1883
COPIED      **                                                              **   1884
COPIED      **  FILE NAME:      CMQGMOV                                     **   1885
COPIED      **                                                              **   1886
COPIED      **  DESCRIPTION:    Get Message Options Structure               **   1887
COPIED      **                                                              **   1888
COPIED      ******************************************************************** 1889
COPIED      **  @START_COPYRIGHT@                                          **    1890
COPIED      **  Statement:     Licensed Materials - Property of IBM         **   1891
COPIED      **                                                              **   1892
COPIED      **                5655-F10                                      **   1893
COPIED      **                (C) Copyright IBM Corporation. 1993, 2002     **   1894
COPIED      **                                                              **   1895
COPIED      **  Status:        Version 5 Release 3                          **   1896
COPIED      **  @END_COPYRIGHT@                                            **    1897
COPIED      ******************************************************************** 1898
COPIED      **                                                              **   1899
COPIED      **  FUNCTION:       This file declares the structure MQGMO,     **   1900
COPIED      **                  which is used by the main MQI.              **   1901
COPIED      **                                                              **   1902
COPIED      **  PROCESSOR:      COBOL                                       **   1903
COPIED      **                                                              **   1904
COPIED      ******************************************************************** 1905
COPIED                                                                           1906
COPIED      **   MQGMO structure                                               1907
COPIED        10 MQGMO.                                                        1908

      COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05          PAGE    40
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED      **    Structure identifier                                           1909
COPIED        15 MQGMO-STRUCID         PIC X(4) VALUE 'GMO '.                  1910
COPIED      **    Structure version number                                       1911
COPIED        15 MQGMO-VERSION         PIC S9(9) BINARY VALUE 1.                 1912
COPIED      **    Options that control the action of MQGET                       1913
COPIED        15 MQGMO-OPTIONS         PIC S9(9) BINARY VALUE 0.                 1914
COPIED      **    Wait interval                                                  1915
COPIED        15 MQGMO-WAITINTERVAL    PIC S9(9) BINARY VALUE 0.                 1916
COPIED      **    Pointer to signal                                              1917
COPIED        15 MQGMO-SIGNAL1         POINTER VALUE NULL.                       1918
COPIED      **    Signal identifier                                              1919
COPIED        15 MQGMO-SIGNAL2         PIC S9(9) BINARY VALUE 0.                 1920
COPIED      **    Resolved name of destination queue                             1921
COPIED        15 MQGMO-RESOLVEDQNAME   PIC X(48) VALUE SPACES.                   1922
COPIED      **    Options controlling selection criteria used for MQGET          1923
COPIED        15 MQGMO-MATCHOPTIONS    PIC S9(9) BINARY VALUE 3.                 1924
COPIED      **    Flag indicating whether message retrieved is in a group        1925
COPIED        15 MQGMO-GROUPSTATUS     PIC X VALUE ' '.                          1926
COPIED      **    Flag indicating whether message retrieved is a segment of a    1927
COPIED      **    logical message                                                1928
COPIED        15 MQGMO-SEGMENTSTATUS   PIC X VALUE ' '.                          1929
COPIED      **    Flag indicating whether further segmentation is allowed for    1930
```

Figure 23 MQPUT Sample Compiled Listing (Page 25 of 32)

```
COPIED      **    the message retrieved                                          1931
COPIED        15 MQGMO-SEGMENTATION   PIC X VALUE ' '.                         1932
COPIED      **    Reserved                                                       1933
COPIED        15 MQGMO-RESERVED1      PIC X VALUE SPACES.                      1934
COPIED      **    Message token                                                  1935
COPIED        15 MQGMO-MSGTOKEN       PIC X(16) VALUE LOW-VALUES.              1936
COPIED      **    Length of message data returned (bytes)                        1937
COPIED        15 MQGMO-RETURNEDLENGTH PIC S9(9) BINARY VALUE -1.               1938
COPIED                                                                         1939
COPIED      ******************************************************************* 1940
COPIED      **  End of CMQGMOV                                               ** 1941
COPIED      ******************************************************************* 1942
      WORKAREA                                                                 1943
      COPY CMQODV COBOL                                                        1944
COPIED      ******************************************************************* 1945
COPIED      **                                                               ** 1946
COPIED      **                WebSphere MQ for z/OS                          ** 1947
COPIED      **                                                               ** 1948
COPIED      **  FILE NAME:      CMQODV                                       ** 1949
COPIED      **                                                               ** 1950
COPIED      **  DESCRIPTION:    Object Descriptor Structure                  ** 1951
COPIED      **                                                               ** 1952
COPIED      ******************************************************************* 1953
COPIED      **  @START_COPYRIGHT@                                            ** 1954
COPIED      **  Statement:     Licensed Materials - Property of IBM          ** 1955
COPIED      **                                                               ** 1956
COPIED      **                5655-F10                                       ** 1957

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    41
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED      **                (C) Copyright IBM Corporation. 1993, 2002    ** 1958
COPIED      **                                                             ** 1959
COPIED      **  Status:       Version 5 Release 3                          ** 1960
COPIED      **  @END_COPYRIGHT@                                            ** 1961
COPIED      ******************************************************************* 1962
COPIED      **                                                               ** 1963
COPIED      **  FUNCTION:       This file declares the structure MQOD,       ** 1964
COPIED      **                  which is used by the main MQI.               ** 1965
COPIED      **                                                               ** 1966
COPIED      **  PROCESSOR:      COBOL                                        ** 1967
COPIED      **                                                               ** 1968
COPIED      ******************************************************************* 1969
COPIED                                                                         1970
COPIED      **  MQOD structure                                                 1971
COPIED        10 MQOD.                                                         1972
COPIED      **    Structure identifier                                         1973
COPIED        15 MQOD-STRUCID           PIC X(4) VALUE 'OD  '.               1974
COPIED      **    Structure version number                                     1975
COPIED        15 MQOD-VERSION           PIC S9(9) BINARY VALUE 1.            1976
COPIED      **    Object type                                                  1977
COPIED        15 MQOD-OBJECTTYPE        PIC S9(9) BINARY VALUE 1.            1978
COPIED      **    Object name                                                  1979
COPIED        15 MQOD-OBJECTNAME        PIC X(48) VALUE SPACES.              1980
COPIED      **    Object queue manager name                                    1981
COPIED        15 MQOD-OBJECTQMGRNAME    PIC X(48) VALUE SPACES.              1982
COPIED      **    Dynamic queue name                                           1983
COPIED        15 MQOD-DYNAMICQNAME      PIC X(48) VALUE 'CSQ.*'.             1984
COPIED      **    Alternate user identifier                                    1985
COPIED        15 MQOD-ALTERNATEUSERID   PIC X(12) VALUE SPACES.              1986
COPIED      **    Number of object records present                             1987
COPIED        15 MQOD-RECSPRESENT       PIC S9(9) BINARY VALUE 0.            1988
COPIED      **    Number of local queues opened successfully                   1989
COPIED        15 MQOD-KNOWNDESTCOUNT    PIC S9(9) BINARY VALUE 0.            1990
COPIED      **    Number of remote queues opened successfully                  1991
COPIED        15 MQOD-UNKNOWNDESTCOUNT  PIC S9(9) BINARY VALUE 0.            1992
COPIED      **    Number of queues that failed to open                         1993
COPIED        15 MQOD-INVALIDDESTCOUNT  PIC S9(9) BINARY VALUE 0.            1994
COPIED      **    Offset of first object record from start of MQOD             1995
COPIED        15 MQOD-OBJECTRECOFFSET   PIC S9(9) BINARY VALUE 0.            1996
COPIED      **    Offset of first response record from start of MQOD           1997
COPIED        15 MQOD-RESPONSERECOFFSET PIC S9(9) BINARY VALUE 0.            1998
COPIED      **    Address of first object record                               1999
COPIED        15 MQOD-OBJECTRECPTR      POINTER VALUE NULL.                  2000
COPIED      **    Address of first response record                             2001
COPIED        15 MQOD-RESPONSERECPTR    POINTER VALUE NULL.                  2002
COPIED      **    Alternate security identifier                                2003
COPIED        15 MQOD-ALTERNATESECURITYID PIC X(40) VALUE LOW-VALUES.         2004
COPIED      **    Resolved queue name                                          2005
COPIED        15 MQOD-RESOLVEDQNAME     PIC X(48) VALUE SPACES.              2006

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    42
      1--------------------VISION:RESULTS FREE FORM TEXT--------------------72          --------
COPIED      **    Resolved queue manager name                                  2007
COPIED        15 MQOD-RESOLVEDQMGRNAME  PIC X(48) VALUE SPACES.              2008
COPIED                                                                         2009
```

Figure 23 MQPUT Sample Compiled Listing (Page 26 of 32)

```
COPIED      ******************************************************************      2010
COPIED      **  End of CMQODV                                               **      2011
COPIED      ******************************************************************      2012
      WORKAREA                                                                      2013
      COPY CMQPMOV COBOL                                                            2014
COPIED      ******************************************************************      2015
COPIED      **                                                              **      2016
COPIED      **                WebSphere MQ for z/OS                         **      2017
COPIED      **                                                              **      2018
COPIED      **  FILE NAME:      CMQPMOV                                     **      2019
COPIED      **                                                              **      2020
COPIED      **  DESCRIPTION:    Put Message Options Structure               **      2021
COPIED      **                                                              **      2022
COPIED      ******************************************************************      2023
COPIED      **  @START_COPYRIGHT@                                           **      2024
COPIED      **  Statement:     Licensed Materials - Property of IBM         **      2025
COPIED      **                                                              **      2026
COPIED      **                5655-F10                                      **      2027
COPIED      **                (C) Copyright IBM Corporation. 1993, 2002     **      2028
COPIED      **                                                              **      2029
COPIED      **  Status:        Version 5 Release 3                         **      2030
COPIED      **  @END_COPYRIGHT@                                             **      2031
COPIED      ******************************************************************      2032
COPIED      **                                                              **      2033
COPIED      **  FUNCTION:       This file declares the structure MQPMO,     **      2034
COPIED      **                  which is used by the main MQI.              **      2035
COPIED      **                                                              **      2036
COPIED      **  PROCESSOR:      COBOL                                       **      2037
COPIED      **                                                              **      2038
COPIED      ******************************************************************      2039
COPIED                                                                              2040
COPIED      **   MQPMO structure                                                    2041
COPIED        10 MQPMO.                                                           2042
COPIED      **    Structure identifier                                            2043
COPIED          15 MQPMO-STRUCID          PIC X(4) VALUE 'PMO '.                  2044
COPIED      **    Structure version number                                        2045
COPIED          15 MQPMO-VERSION          PIC S9(9) BINARY VALUE 1.               2046
COPIED      **    Options that control the action of MQPUT and MQPUT1             2047
COPIED          15 MQPMO-OPTIONS          PIC S9(9) BINARY VALUE 0.               2048
COPIED      **    Reserved                                                        2049
COPIED          15 MQPMO-TIMEOUT          PIC S9(9) BINARY VALUE -1.              2050
COPIED      **    Object handle of input queue                                    2051
COPIED          15 MQPMO-CONTEXT          PIC S9(9) BINARY VALUE 0.               2052
COPIED      **    Reserved                                                        2053
COPIED          15 MQPMO-KNOWNDESTCOUNT   PIC S9(9) BINARY VALUE 0.               2054
COPIED      **    Reserved                                                        2055

      COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    43
      1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
COPIED          15 MQPMO-UNKNOWNDESTCOUNT PIC S9(9) BINARY VALUE 0.               2056
COPIED      **    Reserved                                                        2057
COPIED          15 MQPMO-INVALIDDESTCOUNT PIC S9(9) BINARY VALUE 0.               2058
COPIED      **    Resolved name of destination queue                              2059
COPIED          15 MQPMO-RESOLVEDQNAME    PIC X(48) VALUE SPACES.                 2060
COPIED      **    Resolved name of destination queue manager                      2061
COPIED          15 MQPMO-RESOLVEDQMGRNAME PIC X(48) VALUE SPACES.                 2062
COPIED                                                                            2063
COPIED      ******************************************************************    2064
COPIED      **  End of CMQPMOV                                              **    2065
COPIED      ******************************************************************    2066
      MAIN:                                                                       2067
      * ------------------------------------------------------------ *            2068
      *                                                              *            2069
      *                                                                           2070
      *    Retrieved the values for the following fields in the JCL PARM          2071
      *    statement:                                                             2072
      *      Queue Manager Name                                                   2073
      *      Queue Name                                                           2074
      *      Number of Messages                                                   2075
      *      Message Pad Character                                                2076
      *      Length of Message(s)                                                 2077
      *      Persistence of Message(s)                                            2078
      *                                                                           2079
      *    If no parameters passed to program then                                2080
      *    call USAGE_ERR and exit                                                2081
      *                                                                           2082
           IF DYLPARMLEN EQ 0                                                     2083
              PERFORM USAGE_ERR                                                   2084
              MOVE 8 TO W00_RETURN_CODE                                           2085
              GOTO MAIN_END                                                       2086
           ENDIF                                                                  2087
      *                                                                           2088
      *    Following fields are initialized                                       2089
      *      PARMINFO =  Data retrieved from PARM statement                       2090
      *      INX      =  Index used to find delimiter                             2091
```

Figure 23 MQPUT Sample Compiled Listing (Page 27 of 32)

```
*      INZ      =  Index used to calculate size of PARM value                                  2092
*      INY      =  Index used to determine number of PARMS retrieved                           2093
*                                                                                              2094
*                                                                                              2095
     MOVE DYLPARM    TO PARMINFO                                                              2096
     MOVE 0          TO INX                                                                   2097
     MOVE 0          TO INZ                                                                   2098
     MOVE 1          TO INY                                                                   2099
*                                                                                             2100
*    Separate into the relevant fields any data passed in the                                 2101
*    PARM statement. The fields are separated by a comma. The last field                      2102
*    ends with a blank.                                                                       2103
*                                                                                             2104

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    44
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
     DOWHILE INX LE 60 AND INY LE 6                                                           2105
       IF PARMDATA(INX) EQ ',' OR PARMDATA(INX) EQ ' '                                        2106
         PARMLEN = INX - INZ              ;Length of PARM data value                        2107
         CASE INY                         ;Determine current PARM field                     2108
           WHEN EQ 1                      ;Queue Manager Name?                              2109
            MOVE PARMDATA(INZ)            ;Save Queue Manager Name                          2110
             TO W00_QMGR LENGTH PARMLEN                                                      2111
           WHEN EQ 2                      ;Queue Name?                                      2112
            MOVE PARMDATA(INZ)            ;Save Queue Name                                  2113
             TO W00_QNAME LENGTH PARMLEN                                                     2114
           WHEN EQ 3                      ;Number of Messages?                              2115
            INW = 5 - PARMLEN                                                              2116
            MOVE PARMDATA(INZ)            ;Save Number of Messages                          2117
             TO W00_NUMMSGS_NUM_CHAR(INW) LENGTH PARMLEN                                   2118
           WHEN EQ 4                      ;Message Pad Character?                           2119
            MOVE PARMDATA(INZ)            ;Save Message Pad Character                       2120
             TO W00_PADCHAR LENGTH PARMLEN                                                 2121
           WHEN EQ 5                      ;Length of Message(s)?                            2122
            INW = 5 - PARMLEN                                                              2123
            MOVE PARMDATA(INZ)            ;Save Length of Message(s)                        2124
             TO W00_MSGLENGTH_NUM_CHAR(INW) LENGTH PARMLEN                                 2125
           WHEN EQ 6                      ;Persistence of Message(s)?                       2126
            MOVE PARMDATA(INZ)            ;Save Persistence of Messages                     2127
             TO W00_PERSISTENCE LENGTH PARMLEN                                             2128
         ENDCASE                                                                           2129
         INY = INY + 1                    ;Next PARM data field number                      2130
       ENDIF                                                                               2131
       IF PARMDATA(INX) NE ' '            ;Not the last PARM data field                     2132
         IF PARMDATA(INX) EQ ','          ;End of current PARM data                         2133
           INZ = INX + 1                  ;Start of new PARM data field                     2134
         ENDIF                                                                             2135
         INX = INX + 1                    ;Check next parameter byte                        2136
       ENDIF                                                                               2137
     ENDDO                                                                                 2138
*                                                                                          2139
*    Move the data (spaces if nothing is entered) into the                                 2140
*    relevant API parameter fields                                                         2141
*                                                                                          2142
      MOVE W00_QMGR          TO QMGR                                                       2143
      MOVE W00_MSGLENGTH_NUM TO BUFFERLEN                                                  2144
      MOVE W00_NUMMSGS_NUM   TO W00_NUMMSGS                                                2145
*                                                                                          2146
*    Display parameters to be used in the program                                          2147
*                                                                                          2148
     PRINT   '========================================'                                   2149
     PRINT   'PARAMETERS PASSED :'                                                        2150
     PRINT   '   QMGR       - ', QMGR                                                     2151
     PRINT   '   QNAME      - ', W00_QNAME                                                2152
     PRINT   '   NUMMSGS    - ', W00_NUMMSGS_NUM                                          2153

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    45
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
     PRINT   '   PADCHAR    - ', W00_PADCHAR                                              2154
     PRINT   '   MSGLENGTH  - ', W00_MSGLENGTH_NUM                                        2155
     PRINT   '   PERSISTENCE - ', W00_PERSISTENCE                                         2156
     PRINT   '========================================'                                   2157
*                                                                                          2158
*    Setup the message buffer                                                              2159
*                                                                                          2160
     MOVE 0          TO INX                                                               2161
     MOVE BUFFERLEN  TO INY                                                               2162
*                                                                                          2163
     DOWHILE INX LE INY                                                                   2164
*                                                                                          2165
        MOVE W00_PADCHAR TO BUFFER(INX)                                                   2166
        INX = INX + 1                                                                     2167
*                                                                                          2168
     ENDDO                                                                                2169
*                                                                                          2170
```

Figure 23 MQPUT Sample Compiled Listing (Page 28 of 32)

```
*    Connect to the specified queue manager.                                    2171
*                                                                               2172
     MQCONN    QMGR                  ;Queue Manager Name                       2173
               HCONN                 ;Connection Handle                        2174
               COMPCODE              ;Completion Code                          2175
               REASON                ;Reason Code                              2176
*                                                                               2177
*    If connection failed then display error message                            2178
*    and exit                                                                   2179
*                                                                               2180
     IF COMPCODE NE MQCC_OK                                                    2181
        MOVE 'MQCONN'   TO W00_ERROR_MESSAGE                                   2182
        PERFORM PRINT_MSG                                                      2183
        MOVE REASON     TO W00_RETURN_CODE                                     2184
        GOTO MAIN_END                                                          2185
     ENDIF                                                                     2186
     PRINT  'MQCONN SUCCESSFUL'                                                2187
*                                                                               2188
*                                                                               2189
*    Open the queue for output                                                  2190
*                                                                               2191
     MOVE MQOO_OUTPUT TO HOPTIONS                                              2192
     MOVE W00_QNAME   TO MQOD_OBJECTNAME                                       2193
*                                                                               2194
     MQOPEN    HCONN                 ;Connection Handle                        2195
               MQOD                  ;Object Description                       2196
               HOPTIONS              ;Control Options                          2197
               HOBJ                  ;Object Handle                            2198
               COMPCODE              ;Completion Code                          2199
               REASON                ;Reason Code                              2200
*                                                                               2201
*    If open failed then display error message                                  2202

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    46
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
*    and exit.                                                                  2203
*                                                                               2204
     IF COMPCODE NE MQCC_OK                                                    2205
        MOVE 'MQOPEN'   TO W00_ERROR_MESSAGE                                   2206
        MOVE COMPCODE   TO COMP_CODE                                           2207
        MOVE REASON     TO REASON_CODE                                         2208
        PERFORM PRINT_MSG                                                      2209
        MOVE REASON     TO W00_RETURN_CODE                                     2210
        GOTO DISCONNECT                                                        2211
     ENDIF                                                                     2212
     PRINT  'MQOPEN SUCCESSFUL'                                                2213
*                                                                               2214
*                                                                               2215
*    Set persistence depending on parameter passed                              2216
*                                                                               2217
     IF W00_PERSISTENCE EQ 'P'                                                 2218
        MOVE MQPER_PERSISTENT     TO MQMD_PERSISTENCE                          2219
     ELSE                                                                      2220
        MOVE MQPER_NOT_PERSISTENT TO MQMD_PERSISTENCE                          2221
     ENDIF                                                                     2222
*                                                                               2223
*    Loop until specified number of messages put to queue                       2224
*                                                                               2225
     MOVE 0           TO INX                                                   2226
     MOVE W00_NUMMSGS  TO INY                                                  2227
*                                                                               2228
     DOWHILE INX LT INY                                                        2229
*                                                                               2230
         MOVE MQMI_NONE TO MQMD_MSGID                                          2231
         MOVE MQCI_NONE TO MQMD_CORRELID                                       2232
*                                                                               2233
         MQPUT     HCONN               ;Connection Handle                      2234
                   HOBJ                ;Object Handle                          2235
                   MQMD                ;Messae Descriptor Attributes           2236
                   MQPMO               ;Put Options                            2237
                   BUFFERLEN           ;Message Buffer Length                  2238
                   BUFFER              ;Message Buffer                         2239
                   COMPCODE            ;Completion Code                        2240
                   REASON              ;Reason Code                            2241
*                                                                               2242
*        If put failed then display error message                               2243
*        and break out of loop                                                  2244
*                                                                               2245
         IF COMPCODE NE MQCC_OK                                                2246
            MOVE 'MQPUT'      TO W00_ERROR_MESSAGE                             2247
            PERFORM PRINT_MSG                                                  2248
            MOVE W00_NUMMSGS TO INX                                            2249
            MOVE REASON       TO W00_RETURN_CODE                               2250
         ELSE                                                                  2251
```

Figure 23 MQPUT Sample Compiled Listing (Page 29 of 32)

```
 COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    47
 1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
           PRINT  'MQPUT SUCCESSFUL'                                                          2252
           W00_NUMPUTS = W00_NUMPUTS + 1                                                      2253
           INX         = INX + 1                                                              2254
         ENDIF                                                                                2255
*                                                                                             2256
     ENDDO                                                                                   2257
*                                                                                             2258
*    Display the number of messages successfully put                                          2259
*    to the queue                                                                             2260
*                                                                                             2261
     MOVE W00_NUMPUTS TO NUMBER_PUTS                                                         2262
     PRINT NUMBER_PUTS, ' MESSAGES PUT TO QUEUE'                                             2263
*                                                                                             2264
* Close the queue                                                                             2265
*                                                                                             2266
     MOVE MQCO_NONE TO HOPTIONS                                                              2267
*                                                                                             2268
     MQCLOSE      HCONN               ;Connection Handle                                     2269
                  HOBJ                ;Object Handle                                         2270
                  HOPTIONS            ;Control Options                                       2271
                  COMPCODE            ;Completion Code                                       2272
                  REASON              ;Reason Code                                           2273
*                                                                                             2274
*    Test the output of the MQCLOSE call.  If the call failed,                                2275
*    print an error message                                                                   2276
*                                                                                             2277
     IF COMPCODE NE MQCC_OK                                                                  2278
        MOVE 'MQCLOSE'  TO W00_ERROR_MESSAGE                                                 2279
        PERFORM PRINT_MSG                                                                    2280
        MOVE REASON TO W00_RETURN_CODE                                                       2281
     ELSE                                                                                    2282
        PRINT 'MQCLOSE SUCCESSFUL'                                                           2283
     ENDIF                                                                                   2284
*                                                                                             2285
DISCONNECT:                                                                                  2286
*                                                                                             2287
* Disconnect from the queue manager                                                           2288
*                                                                                             2289
     MQDISC    HCONN                  ;Connection Handle                                     2290
               COMPCODE               ;Completion Code                                       2291
               REASON                 ;Reason Code                                           2292
*                                                                                             2293
*    Test the output of the disconnect call.  If the call failed,                             2294
*    print an error message                                                                   2295
*                                                                                             2296
     IF COMPCODE NE MQCC_OK                                                                  2297
        MOVE 'MQDISC'   TO W00_ERROR_MESSAGE                                                 2298
        PERFORM PRINT_MSG                                                                    2299
        MOVE REASON     TO W00_RETURN_CODE                                                   2300

COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    48
1-------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
     ELSE                                                                                    2301
        PRINT 'MQDISC SUCCESSFUL'                                                            2302
     ENDIF                                                                                   2303
*                                                                                             2304
MAIN_END:                                                                                    2305
*                                                                                             2306
*    Set the return code                                                                      2307
*                                                                                             2308
     MOVE W00_RETURN_CODE TO DYLRETURN                                                       2309
*                                                                                             2310
     STOP                                                                                    2311
*                                                                                             2312
* ------------------------------------------------------------ *                             2313
USAGE_ERR:                                                                                   2314
* ------------------------------------------------------------ *                             2315
*                                                                                             2316
     PRINT '=============================================='                                  2317
     PRINT 'PARAMETERS FOR PROGRAM :'                                                        2318
     PRINT '     QMGR        - QUEUE MANAGER'                                                2319
     PRINT '     QNAME       - QUEUE NAME'                                                   2320
     PRINT '     NUMMSGS     - NUMBER OF MESSAGES'                                           2321
     PRINT '     PADCHAR     - MESSAGE PADDING CHARACTER'                                    2322
     PRINT '     MSGLENGTH   - LENGTH OF MESSAGE(S)'                                         2323
     PRINT '     PERSISTENCE - PERSISTENCE OF MESSAGE(S)'                                    2324
     PRINT '=============================================='                                  2325
*                                                                                             2326
*                                                                                             2327
* ------------------------------------------------------------ *                             2328
PRINT_MSG:                                                                                   2329
* ------------------------------------------------------------ *                             2330
*                                                                                             2331
```

Figure 23 MQPUT Sample Compiled Listing (Page 30 of 32)

```
      MOVE COMPCODE   TO COMP_CODE                                                          2332
      MOVE REASON     TO REASON_CODE                                                        2333
      PRINT '*************************************************'                            2334
      PRINT '* ', W00_ERROR_MESSAGE                                                         2335
      PRINT '* COMPLETION CODE : ', COMP_CODE                                               2336
      PRINT '* REASON CODE     : ', REASON_CODE                                             2337
      PRINT '*************************************************'                            2338

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE   49
           C R O S S   R E F E R E N C E
                                 A       T D
                                 R       Y E
         DATANAME                E       P C
          OR TAG             LOCN A  SIZE  E . DEFN            REFERENCES

BUFFER                       1177 P 32767 CH    184 -  2166 2234
BUFFERLEN                    1169 P     4 BI 0  182 -  2144 2162 2234
COMP_CODE                     595 O     4 CH    105 -  2207 2332 2336
COMPCODE                      953 P     4 BI 0  176 -  2173 2181 2195 2205 2207 2234
                                                       2246 2269 2278 2290 2297 2332
DISCONNECT:                                    2286 -  2211
DYLPARM                        41 U    60 CH    DYL -  2096
DYLPARMLEN                    364 O     2 PD    DYL -  2083
DYLRETURN                    RTC  U     2 BI    DYL -  2309
HCONN                         929 P     4 BI 0  170 -  2173 2195 2234 2269 2290
HOBJ                          937 P     4 BI 0  172 -  2195 2234 2269
HOPTIONS                      945 P     4 BI 0  174 -  2192 2195 2267 2269
INW                          INW  U     2 BI    DYL -  2116 2123
INX                          INX  U     2 BI    DYL -  2097 2105 2107 2134 2136 2136
                                                       2161 2164 2167 2167 2226 2229
                                                       2249 2254 2254
INY                          INY  U     2 BI    DYL -  2099 2105 2108 2130 2130 2162
                                                       2164 2227 2229
INZ                          INZ  U     2 BI    DYL -  2098 2107 2134
LIT017001                    7241 R    17 CH   2151 -  2152 2153 2154 2155 2156
LIT020001                    8958 R    20 CH   2336 -  2337
MAIN_END:                                      2305 -  2086 2185
MQCC_OK                      4222 R     4 BI 0 1172 -  2181 2205 2246 2278 2297
MQCI_NONE                    2543 R    24 CH    620 -  2232
MQCO_NONE                    3214 R     4 BI 0  889 -  2267
MQMD                         6289 R   364 CH   1815 -  2234
MQMD_CORRELID                6361 R    24 CH   1841 -  2232
MQMD_MSGID                   6337 R    24 CH   1839 -  2231
MQMD_PERSISTENCE             6333 R     4 BI 0 1837 -  2219 2221
MQMI_NONE                    2519 R    24 CH    617 -  2231
MQOD                         6761 R   336 CH   1972 -  2195
MQOD_OBJECTNAME              6773 R    48 CH   1980 -  2193
MQOO_OUTPUT                  3782 R     4 BI 0 1051 -  2192
MQPER_NOT_PERSISTENT         2507 R     4 BI 0  612 -  2221
MQPER_PERSISTENT             2511 R     4 BI 0  613 -  2219
MQPMO                        7097 R   128 CH   2042 -  2234
NUMBER_PUTS                   603 O     4 CH    107 -  2262 2263
PARMDATA                      609 O     1 CH    110 -  2106 2106 2110 2113 2117 2120
                                                       2124 2127 2132 2133
PARMINFO                      609 O    60 CH    109 -  2096
PARMLEN                       545 O     2 BI 0  103 -  2107 2110 2113 2116 2117 2120
                                                       2123 2124 2127
PRINT_MSG:                                     2329 -  2183 2209 2248 2280 2299

COMPUTER ASSOCIATES      VISION:RESULTS   6.0 *             DATE  09/20/05           PAGE   50
           C R O S S   R E F E R E N C E
                                 A       T D
                                 R       Y E
         DATANAME                E       P C
          OR TAG             LOCN A  SIZE  E . DEFN            REFERENCES

QMGR                          881 P    48 CH    168 -  2143 2151 2173
REASON                        961 P     4 BI 0  178 -  2173 2184 2195 2208 2210 2234
                                                       2250 2269 2281 2290 2300 2333
REASON_CODE                   599 O     4 CH    106 -  2208 2333 2337
USAGE_ERR:                                     2314 -  2084
W00_ERROR_MESSAGE             689 O    48 CH    119 -  2182 2206 2247 2279 2298 2335
W00_MSGLENGTH_NUM             865 P     5 NU 0  134 -  2144 2155
W00_MSGLENGTH_NUM_CHAR        865 P     5 CH    135 -  2124
W00_NUMMSGS                   857 P     4 BI 0  133 -  2145 2227 2249
W00_NUMMSGS_NUM               849 P     5 NU 0  130 -  2145 2153
W00_NUMMSGS_NUM_CHAR          849 P     5 CH    131 -  2117
W00_NUMPUTS                   681 O     4 BI 0  118 -  2253 2253 2262
W00_PADCHAR                   833 O     1 CH    125 -  2120 2154 2166
W00_PERSISTENCE               873 P     1 CH    137 -  2127 2156 2218
W00_QMGR                      737 O    48 CH    123 -  2110 2143
W00_QNAME                     785 O    48 CH    124 -  2113 2152 2193
W00_RETURN_CODE               673 O     2 BI 0  117 -  2085 2184 2210 2250 2281 2300
                                                       2309
```

Figure 23   MQPUT Sample Compiled Listing (Page 31 of 32)

```
COMPUTER ASSOCIATES       VISION:RESULTS   6.0 *              DATE  09/20/05            PAGE    51
1--------------------VISION:RESULTS FREE FORM TEXT--------------------72         --------
DEFAULT OPTIONS SPECIFIED ARE:

   ENVIRONMENT IS MVS
   CPUID  2084       C0851

   ASALINE(N/), BATCHIQ(IIBATCH/), CBXSIGN(N/), CENTINEW(75/), CENTRY1(/), CENTRY2(/), COBEDIT(A/),
   COBENV(N/), COMPERR(N/), COB2NR(N/), COMPWRK(1500K/), CURNCY($/), DATATRN(N/), DECIML9(E/),
   DELIM(/), DLMFRST(Y/), DUPCENM(N/), DYLVARP(N/), DYL4YR(N/), EDP1ZERO(N/), EDSUPR(N/),
   EURODAT(N/), EURONUM(N/), EXCEL(N/), EXCLPAT(N/), EXPRERR(N/), FREEMEM(1000K/),
   FREEZDD(SYS280FZ/), GETMAX(2500K/), KWDLIT(N/), LE(Y/), LIBRBUF(60K/), LPPUNLMT(N/), LSTSTMX(N/),
   LITRFROM(N/), LITRZERO(N/), MACHORG(N/), MAXDNLN(50/), MAXDYLF(175/), MNAMENU(N/), NAMEHDR(/),
   NDVRCOM(N/), NDVRENV(/), NODLETE(N/), NOVSOIO(N/), NOPOWRT(N/), NOSRTAB(N/), NOTOTAL(N/),
   NOVSIO(N/), NUMCHAR(N/), NUMPD(N/), OPTPRDG(Y/), OPTPRER(N/), OUTFILE(N/), PANVBUF(60K/),
   PDSREPL(N/), PGLINER(55/), PGLINES(55/), PROGMOD(XREFREF,CONVENTIONAL,EXP/,STRUCT2), PRTCTRS(N/),
   PRTZERO(N/), QLF(N/), RANDMPCT(N/), RDYONLY(N/), RESRWRD(N/), RETCODE(Y/), RPTASA(N/),
   RPTDDNM(SYS280R/), RPTXPAG(N/), SORTDEV(SYSDA/), SORTDYN(N/), SORTMEM(36K,100K,700K,100K/),
   SORTNAM(SORT/), SSMASK(N/), STRUCGO(N/), SUBRADD(Y/), SUPCOBW(Y/), SUPRESQ(N/), SUP182W(N/),
   SYSBLOK(N/), TIMESEP(D/), TABLEHI(N/), VSAMCAT(Y/), VDUPABND(N/), VSAMMSG(N/), XREF$(N/),
   ZDIVAB(O/)

COMPUTER ASSOCIATES       VISION:RESULTS  6.0 *              DATE  09/20/05  18.16.49   PAGE     1
        1        2        3        4        5        6        7        8        9
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
==========================================
PARAMETERS PASSED :
   QMGR       - CSQ1
   QNAME      - VISION.RESULTS.TEST.QUEUE
   NUMMSGS    - 00001
   PADCHAR    - A
   MSGLENGTH  - 00040
   PERSISTENCE - N
==========================================
MQCONN SUCCESSFUL
MQOPEN SUCCESSFUL
MQPUT SUCCESSFUL
0001 MESSAGES PUT TO QUEUE
MQCLOSE SUCCESSFUL
MQDISC SUCCESSFUL
        1        2        3        4        5        6        7        8        9
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
COMPUTER ASSOCIATES       VISION:RESULTS  6.0 *              DATE  09/20/05  18.16.49   PAGE     2
        1        2        3        4        5        6        7        8        9
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
VISION:RESULTS CONTROL TOTALS

FILE       RECORD         CHARACTER     BLOCK     DROPPED     REWRITTEN      INSERTED      ERASED
 ID         COUNT           COUNT       COUNT   BLOCK COUNT  RECORD COUNT  RECORD COUNT RECORD COUNT


                          RECORDS                 PAGES

FILE PRINT                        15                  2

REPORT PRINT

FIXED BLANK COUNT

FIXED DECIMAL DIVIDE


    RETURN CODE-0000


   ******************************************************
   *                                                    *
   *  (C) 2005 COMPUTER ASSOCIATES INTERNATIONAL, INC.  *
   *                                                    *
   ******************************************************
```

Figure 23 MQPUT Sample Compiled Listing (Page 32 of 32)

Chapter 5: CSV Utility

The comma-separated values (CSV) utility transforms the data in your VISION:Results program into and from CSV-formatted data. CSV produces data in a format common to most PC spreadsheets, databases, and files used on applications for the import and export of data. CSV can either convert a field of any type and length to a CSV format or convert a CSV-formatted field back to a VISION:Results data field.

The field value is converted into character format, enclosed by a pair of quotation mark symbols, with a delimiting symbol appended to the end of the quoted character value. You can designate replacement symbols for the quotation mark, delimiter, and other editing symbols.

An application calls the CSV utility repeatedly for each field that is to be converted and placed into the user's buffer. When all the fields have been placed into the buffer, the application writes the buffer to an output file for later downloading to the PC application. The same repetitive process is used for the conversion of fields from a CSV-formatted file to that of a file containing VISION:Results data types.

LE must be enabled either via the OPTION statement or DYLINSTL LE=Y to use this feature.

Use of Symbols

Any symbol can be used to delimit, enclose, or punctuate the field. For this discussion, the most commonly specified values are used.

Example

The value 48000 is first shown in numeric form, then as a CSV-formatted field:

Numeric value	48000
CSV format	“48000”,

If the converted field contains a character that duplicates either the quotation mark character or the escape character, CSV precedes the character in the converted field with an escape character.

Example

Field value	Reply with a “Y” to continue.
CSV format	“Reply with a \“Y\” to continue.”,
Field value	Precede the directory name with a \.
CSV format	“Precede the directory name with a \\.”,

Numeric Fields

A numeric field can be specified as binary, packed, or zoned. When these values are converted to character, all leading zeros are suppressed. The decimal point symbol is inserted if the specified scale is greater than zero. A leading sign symbol is inserted if the user-designated symbol for the appropriate sign (positive or negative) is not blank. If it is blank, no sign symbol is inserted. If the value of the numeric field is zero, then the converted field is a single character of zero.

Example

Field value	000059	scale=0	positive value (positive sign symbol is blank)
CSV format	“59”,		
Field value	45008	scale=3	positive value (positive sign symbol is blank)
CSV format	“45.008”,		
Field value	0092	scale=3	negative value (negative sign symbol = –)
CSV format	“-.092”,		
Field value	20000	scale=1	positive value (positive sign symbol = +)
CSV format	“+2000.0”,		
Field value	0		
CSV format	“0”,		
Field value	0	scale=2	
CSV format	“.00,”		

Floating Point Value

A floating point value is always converted to character in the familiar exponential format: n.nnnnnnE+xx.

The numeric portion of a floating point value is always represented in normalized form, one integer followed by six decimal place integers. The numeric portion contains a leading sign if the value is negative. Otherwise, no sign is appended. The exponent follows the character E. The exponent is always two digits and is always preceded by a sign for both positive and negative exponents. The user-designated symbols have no effect on the format of a floating point value.

Example

Here are some examples of floating point values in CSV format:

"5.000000E+02",

"-9.345678E-01",

"1.999900E+18",

CSVRSLT API Description

All access to the CSV utility is handled through a C environment Application Programmer Interface (API) named CSVRSLT. The CSV utility runs in 31-bit mode, above the 16-megabyte line. The addressing mode (AMODE) is 31. The residency mode (RMODE) is ANY. The CSVRSLT API consists of 12 required parameters. The first nine are input parameters that you define at the time of the call. They provide the function to be performed, the characteristics of the field to be converted or created, and the characteristics of the CSV record or buffer that is used either to write or to read the converted value. The remaining three are return parameters.

The first parameter is a function code indicating the type of operation consisting of one of the following:

TOCSV	Converts a field from a specified type and length to character format and places it in the buffer in CSV format. The field is delimited by user-designated symbols (usually a comma) and enclosed within a pair of user-designated symbols (usually double quotation marks).
FROMCSV	Converts a CSV-formatted field and places it into an application field. The field is in the proper format to be used by the application.
TERM	Indicates that this is the final call to CSV and the C environment is terminated.

Parameters two through five are the application field parameters that are used to provide the characteristics of the field which are either input to create a CSV record or buffer (TOCSV function), or are the created output from a CSV record or buffer (FROMCSV function). These four application field parameters consist of the type of field, length of field, scale of field, and the field itself.

Parameters six through nine are the CSV buffer parameters that are used to provide the characteristics related to accessing or creating the CSV record or buffer. These parameters provide the address of the buffer, length of the buffer, the field's position or offset in the buffer, and symbols used to format the field.

Parameters 10 through 12 are returned by the CSVRSLT API and provide the length of the converted field, a return code, and a reason code.

Parameters for the TOCSV and FROMCSV Functions

The TOCSV and FROMCSV functions use the same 12 parameters. When the TOCSV function is the first parameter, the application field parameters (parameters two through five) provide the input information and the CSV buffer parameters (parameters six through nine) provide the output information to create a CSV-formatted value field. When the FROMCSV function is the first parameter, the application field parameters provide the output information and the CSV buffer parameters provide the input information to populate a VISION:Results data field.

The format of the CSVRSLT call and detailed coding of the parameters follows.

Syntax

```
CALL CSVRSLT [CDLOAD] USING    fcncode, fldtype, fldlength, fldscale,
                               fldvalue, bufptr, bufsize, bufoffset,
                               usersymbols, cnvlength, retcode,
                               reascode
```

Commas separating parameters are optional.

Note: Use CDLOAD or 90K for VSE only.

The following input parameters are 4-byte addresses:

fcncode — 8-byte area, in character format, containing the value of either TOCSV or FROMCSV. Note this value must be padded with trailing blanks to formulate an 8-byte value.

fldtype — 2-byte area, in character format, containing the field indicator and type. This can be created using the CALL statement parameter T'dataname (for example, if the data name is BALANCE, code T'BALANCE). If a VISION:Results field is not being used, the contents of the parameter are as follows:

Byte 1, field type indicator: T

	Byte 2, field type:	B	binary
		C	character
		N	numeric (zoned format)
		P	packed

fldlength — 3-byte area. This can be created using the CALL statement parameter L'dataname (for example, if the data name is BALANCE, code L'BALANCE). If a VISION:Results field is not being used, the contents of the parameter are as follows:

Byte 1, field length indicator in character format:	L	
Bytes 2-3, field length in binary format: (Allowable lengths depend on the field type.)	B	1 to 4
	C	0 to 32,767
	N	1 to 16
	P	1 to 16

Note that a length of zero is allowed for a character field. This creates a null value in the buffer consisting of a pair of quotation mark symbols followed by the delimiter symbol ("",).

fldscale — 2-byte area. This can be created using the CALL statement parameter D'dataname (for example, if the data name is BALANCE, code D'BALANCE). If a VISION:Results field is not being used, the contents of the parameter are as follows:

Byte 1, field scale indicator in character format:	D

Byte 2, scale of the field (number of decimal places) in binary format:

Allowable values depend on the field type.

Field Type	Field Length	Minimum Scale	Maximum Scale
B	1	0	3
	2	0	5
	3	0	7
	4	0	10
C		0	0
N		0	fldlength
P		0	(fldlength*2)-1 to a maximum of 16

fldvalue	The field or data name to be converted or created. The previous parameters, fldtype, fldlength, and fldscale describe the characteristics of this parameter.
bufptr	Address of the beginning of the file buffer that holds the converted field values. If the converted value does not fit within the confines described by the buffer size and offset, the buffer is cleared to blanks starting from the offset to the end of the buffer.
bufsize	4-byte area, in binary format, containing the length of the buffer for the converted field value. The buffer size must be greater than zero.
bufoffset	4-byte area, in binary format, containing the offset for the buffer, relative to zero, at which to place the converted value. The buffer offset must be greater than or equal to zero and less than the buffer size.
usersymbol	8-byte area, in character format, containing user-designated symbols. These symbols are used when quoting, delimiting, and punctuating fields when converting them into CSV format. The symbols are defined in the following order: quotation mark, delimiter, escape, decimal point, positive sign, negative sign, quote numeric option flag, unused. See User-Designated Symbols on page 138 for more information about user-designated symbols.

The following return parameters are 4-byte addresses:

cnvlength	4-byte area, in binary format, containing the length of the converted field and reflecting the actual amount of space occupied by the converted field in the buffer. This includes the length of the field (in character format with all necessary punctuation) plus the delimiting symbol and the pair of symbols used to enclose the value.
retcode	4-byte area, in binary format, containing the return code. See Return Codes and Reason Codes on page 143 for more information.
reascode	4-byte area, in binary format, containing the reason code. See Return Codes and Reason Codes on page 143 for more information.

User-Designated Symbols

User-designated symbols are used when quoting, delimiting, and punctuating fields defined as input parameters to the CSV utility. Except for the quote numeric option flag, there are no default values; each symbol must be specified.

These symbols are inserted into the field value after it has been converted to character format. The symbols are discussed in the following order:

- Quotation mark
- Delimiter
- Escape
- Decimal point
- Positive sign
- Negative sign
- Quote numeric option flag
- Unused

Quotation Mark Symbol

The quotation mark symbol is used to enclose the value. It is inserted at the beginning and the end of the value. It is typically specified as double quotation marks (“).

Example: “900.3”

Delimiter Symbol

The delimiter symbol is used to delimit the value from any value that follows it in the buffer. It is appended to the end of the value after the added quotation marks. It is typically specified as a comma (,).

Example: “California”,

Escape Symbol

The escape symbol is used when a value contains a character that is the same as the quotation mark symbol or the escape symbol. When this duplication occurs, the escape symbol is inserted into the value preceding the character. It is typically specified as a backslash (\).

Example: “Reply with a \“Y\” to continue.”,

Decimal Point Symbol

The decimal point symbol is used when a numeric field (type B, N, P, or Z) has a non-zero scale. The decimal point symbol is inserted into the value at the relevant position to indicate its scale. It is typically specified as a period (.).

Example: “4100.50”,
Example if symbol is , (comma) : “4100,50”,

This specification has no effect on the format of a floating point number (type E). A decimal point is always indicated with a period (.), regardless of the symbol designated.

Example if symbol is , (comma) : “5.999712E+02”,

Positive Sign Symbol

The positive sign symbol is used when a numeric field (type B, N, P, or Z) has a value greater than or equal to zero. If the symbol is not blank, it is inserted into the value prior to the first digit. If the symbol is blank, nothing is inserted. It is typically specified as a blank. If not blank, it is typically specified as a plus sign (+).

```
Example if symbol is blank:     “625”,
Example if symbol is + :        “+625”,
```

This specification has no effect on the format of a floating point number (type E). A positive sign is always indicated with a plus sign (+), regardless of the symbol designated.

```
Example if symbol is #          “5.999712E+02”,
(pound sign) :
```

Negative Sign Symbol

The negative sign symbol is used when a numeric field (type B, N, P, or Z) has a value less than zero. If the symbol is not blank, it is inserted into the value prior to the first digit. If the symbol is blank, nothing is inserted. If the symbol is either a left parenthesis (() or a right parenthesis ()), the value is bracketed with a pair of parentheses. It is typically specified as a minus sign (-).

```
Example if symbol is blank:     “625”,
Example if symbol is - :        “-625”,
Example if symbol is ( :        “(625)”,
```

This specification has no effect on the format of a floating point number (type E). A negative sign is always indicated with a minus sign (-) regardless of the symbol designated.

```
Example if symbol is #          “-5.999712E-02”,
(pound sign) :
```

Quote Numeric Option Flag

The quote numeric option flag determines whether a numeric field should be encapsulated with the quotation symbol when moved to the CSV buffer during a TOCSV operation. The numeric field will be encapsulated if the flag is set to blank or Y. If the quote numeric option flag is set to N, CSVRSLT will not encapsulate numeric fields.

```
Example if flag is blank or Y:  “625”,
Example if flag is N:           625,
```

Unused

The eighth byte of the user symbol area is reserved.

Parameters for Terminating CSV

A function code with a value of TERM terminates the C environment when no more calls are to be made to the CSV utility.

This function code requires the following parameter to be passed to CSVRSLT.

Syntax

```
CALL CSVRSLT [CDLOAD]* USING fcncode
```

Note: Use CDLOAD or 90K for VSE only.

The following input parameter is a 4-byte address.

fcncode	8-byte area, in character format, containing the value TERM. This value must be padded with trailing blanks to formulate an 8-byte value.

Using the API

An application typically calls the CSV utility repeatedly for each converted field.

TOCSV

The TOCSV function causes each call to the CSV utility to append the field to the end of the data that resides in the buffer from previous calls. To accomplish this, elect to increment either the buffer address or the buffer offset by the value passed back in the converted field length return parameter (cnvlength). Each subsequent call to the CSV utility uses this incremented value as an input parameter, effectively causing the field to be appended to the buffer.

When the application has all the fields in the buffer it typically writes the buffer to an output file. Prior to this some final processing is necessary. Every field placed in the buffer, including the last field, is followed by a delimiter. It is the application's responsibility to remove the delimiter from the last field. If you want the buffer to terminate with a carriage return and line feed (CR and LF), it is your responsibility to append the appropriate characters to the end of the buffer.

You can place a null value in the buffer under certain conditions. This is accomplished by calling the CSV utility with parameters specifying a character field with a length of zero. This causes a pair of quotation mark symbols and a delimiter symbol to be placed in the buffer.

FROMCSV

Before the FROMCSV function is used, the CSV-formatted data or string must be read or moved into a VISION:Results WORKAREA field or a FILE record. For example, a FILE statement and a READ statement reads a CSV record or string into a FILE record; or a user subroutine is called and the WORKAREA field is specified as a parameter and the subroutine moves the CSV string into the WORKAREA field.

Once the CSV string has been read into the program, move FROMCSV into the first parameter (fcncode). Next, the buffer location (name of the WORKAREA field or FILE record), the buffer size of the CSV string (size of the WORKAREA field or FILE record), and the CSV buffer parameters must be placed in the parameter list (bufptr, bufsize, and bufoffset). The initialization of these parameters is done once for each CSV record. Do not initialize them again until you read another CSV record.

The application data parameter fields (fldtype, fldlength, fldscale, and fldvalue) are coded in the same manner as the TOCSV function. These attribute field parameters are used in retrieving the CSV converted field from the CSV record or string. An error is returned by CSVRSLT in the following instances:

- The CSV converted field length is greater than the specified field length parameter (fldlength)
- A closing quote symbol is missing from the CSV string
- The scale of the CSV converted field is different than the field scale parameter (fldscale)

After each CSVRSLT call, the converted field length return parameter (cnvlength) contains the length of the CSV converted field or token prior to conversion. The converted value is placed into the field parameter (fldvalue). The buffer offset parameter (bufoffset) is also automatically incremented by the converted field length parameter so that the buffer offset parameter is pointing to the next CSV field to be converted.

To retrieve the next value from the CSV buffer, specify the attributes of the next application data parameter fields (fldtype, fldlength, fldscale, and fldvalue). To avoid an error status being returned, these parameters need to coincide with the value from the CSV buffer

When all fields have been retrieved from a CSV record, the next record must be read into storage and the above sequence repeated.

TERM

When all processing is complete, make a final call to the CSV utility with a function code of TERM. This terminates the C environment. Although this is not a requirement, it is recommended in order for the CSV utility to perform all necessary memory management, clean up, and close files. Failure to make the termination call can result in missing error and diagnostic messages, or other unpredictable results.

Error Codes

This section describes the types of error codes generated by the CSV utility.

Return Codes and Reason Codes

The following table lists the return and reason codes that the CSV utility passes back in the retcode parameter. Each return code has a set of reason codes associated with it. The reason codes are set in the reascode parameter.

Return Code	Reason Code		Description
	Dec	Hex	
0			The field converted successfully.
4			An error occurred.
	1	X'01'	The function code was not recognized. The function code must be either TOCSV, FROMCSV, or TERM. Note that each value must be padded with blanks to formulate an 8-byte value.
	2	X'02'	The field type was not recognized. The field type must be B, C, E, P, or Z.
	3	X'03'	The length of the binary field was not within the allowable limits. The length of a binary field must be between 1 and 4.
	4	X'04'	The scale of the binary field was not within the allowable limits. The scale of a binary field must be between 0 and the largest scale allowed for the specified field length. Length — Maximum Scale 1 — 3 2 — 5 3 — 7 4 — 10
	5	X'05'	The binary field did not convert to character successfully (TOCSV). The binary field does not contain a valid digit (FROMCSV).

Return Code	Reason Code		Description
	Dec	Hex	
	6	X'06'	The length of the character field was not within the allowable limits. The length of a character field must be between 0 and 32767 (TOCSV). The length of the character field was not within the range of the CSV-converted field (FROMCSV).
	7	X'07'	The scale of the character field was not zero. The scale of the CSV-converted field is different than the field scale parameter (FROMCSV).
	8	X'08'	The length of the floating point field was not within the allowable limits. The length of a floating point field must be 4.
	9	X'09'	The scale of the floating point field was not zero. The scale of the CSV-converted field is different than the field scale parameter (FROMCSV).
	10	X'0A'	The floating point field did not convert to character successfully.
	11	X'0B'	The length of the packed field was not within the allowable limits. The length of a packed field must be between 1 and 16 (TOCSV). The CSV-converted field was not within the allowable limits. The CSV-converted field must be between 1 and 30, inclusive (CSVFROM).
	12	X'0C'	The scale of the packed field was not within the allowable limits. The scale of a packed field must be between 0 and twice the specified field length minus 1, up to a maximum scale of 16.
	13	X'0D'	The packed field did not convert to character successfully. It did not consist of valid digits or did not contain a valid packed sign.
	14	X'0E'	The length of the zoned field was not within the allowable limits. The length of a zoned field must be between 1 and 16 (TOCSV). The length of the zoned field was not within the allowable limits. The CSV-converted field length must be greater than the zoned field length (FROMCSV).

Return Code	Reason Code		Description
	Dec	Hex	
	15	X'0F'	The scale of the zoned field was not within the allowable limits. The scale of a zoned field must be between 0 and the specified field length.
	16	X'10'	The zoned field did not convert to character successfully. It did not consist of valid digits or did not contain a valid sign and digit combination in the last position.
	17	X'11'	The CSV-converted field or string that was read as input was missing either the leading or closing encapsulated quote (FROMCSV).
	18	X'12'	No data was found for the CSV-converted field or string that was read as input (CSVFROM).
	19	X'13'	The quote numeric option flag was set to N and the field was not numeric (TOCSV).
8			A severe error occurred.
	1	X'01'	There was an error in the buffer offset or the size parameter. Either the buffer size was less than or equal to zero, the buffer offset was less than zero, or the buffer offset was greater than or equal to the buffer size (TOCSV).
	2	X'02'	The length of the user buffer was exceeded. The converted field, including all punctuation, did not fit within the buffer provided (TOCSV). Note that if the converted value does not fit within the confines described by the buffer size and offset, the buffer is cleared to blanks starting from the offset to the end of the buffer. Any data that is at that location upon input is cleared to blanks.

Return Code	Reason Code		Description
	Dec	Hex	
not set	not set	not set	Incorrect number of parameters. An error condition occurs when an incorrect number of parameters is specified. There must be exactly 12 parameters specified. Or, if the function code is 'TERM ' , then either 1 parameter or 12 parameters must be specified. If the CSV utility detects that the number of parameters is incorrect, it sets a value of 12 into register 15 and returns immediately, without performing any processing. Because there was an error in the parameter list itself, the return code and reason code parameters cannot be set by the CSV utility. The only indication of this error is the return value in register 15. To view the contents of register 15, you must set the RETCODE parameter in the DYINSTL macro to Y. For more information, see the *Advantage VISION:Results for z/OS Installation Guide*.

Condition Code 0077

0077 is the completion code used to indicate any CSV error detected by VISION:Results when the program is using the recommended Copy Support Routines (see Copy Support Routines on page 162).

To maintain this common VISION:Results standard of COND CODE 0077, if the Copy Support Routines are not used, then set the COND CODE to 0077 in DYLRETURN (see the MOVE 77 TO DYLRETURN statement in Figure 24). Always supply significant information needed to isolate the issue. The data supplied by the HEXPRINT and display statements in the sample shows information needed to isolate the issue.

```
IF BINFIELD NE 0
       HEXPRINT 'REGISTER 15 = '     BINFIELD
                'CSV RETURN CODE = ' CSVRETCD
                'CSV REASON CODE = ' CSVRSNCD
       HEXPRINT 'NAME = '   NAME
       HEXPRINT 'CSVBUF = ' CSVBUF
       MOVE 77 TO DYLRETURN
       STOP
    ENDIF
```

Figure 24 Sample of Condition Code 0077

Sample Program Using TOCSV Function for z/OS

```
;=================================================================
; THIS PROGRAM USES THE CSV SYSTEM'S TOCSV FUNCTION TO CREATE CSV
; FORMATTED RECORDS. THE PROGRAM READS THE ACCOUNTS RECEIVABLE SAMPLE
; FILE AND CONVERTS THE NAME, ACCOUNT, AND BALANCE FIELDS OF CERTAIN
; RECORDS TO CSV FORMATTED RECORDS. THE CSV FORMATTED RECORDS WILL BE
; PRINTED AS A REPORT AND WRITTEN TO A NEW VARIABLE SEQUENTIAL DATASET.
; THE VARIABLE SEQUENTIAL DATASET WILL BE USED AS INPUT BY THE SAMPLE
; PROGRAM USING THE FROMCSV FUNCTION.
; NOTE:  FOR DYLCOMRG TO PICK UP THE VALUE IN REGISTER 15, PLEASE SET
; RETCODE=Y IN THE DYLINSTL MACRO. FOR MORE INFORMATION, SEE THE SECTION
; "OPTIONAL PARAMETERS" IN THE VISION:RESULTS INSTALLATION GUIDE
;=================================================================
OPTION STRUCTURED2
FILE ARFILE FB 352  STATUS ARSTAT
ACCOUNT  7   4 NU     (ACCOUNT'NUMBER)
NAME    25  85 CH     (CUSTOMER'NAME)
BALANCE  5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE 2 182 CH     (ACCOUNT'CODE)

FILE CSVFILE VB 120 OUTPUT FROM CSVFILE LENGTH CSVLEN
CSVSTRING   120

WORKAREA
FUNCCODE   8 CH

CSVBUFSZ   4 BI
CSVBUFOF   4 BI
USRSYMBL   8 CH
CONVLTH    4 BI
CSVRETCD   4 BI
CSVRSNCD   4 BI
CSVBUF   100 CH (CSV BUFFER)

WORKAREA
 CHFIELD     2  CH
  REDEFINE CHFIELD
 BINFIELD    2  BI

REPORT1 120 WIDE

ON ONE
MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 'TOCSV ' TO FUNCCODE
MOVE 120 TO CSVBUFSZ
MOVE 0 TO CSVBUFOF
MOVE 0 TO CONVLTH
ENDONE

IF (ACCTCODE EQ 'MA') AND (CSVRETCD EQ 0)

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

;=================================================================
; CONVERT THE NAME FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE NAME FIELD IS PLACED AT THE BEGINNING OF THE BUFFER.
;=================================================================

; SET THE OFFSET TO ZERO TO PLACE THE NAME FIELD
; AT THE BEGINNING OF THE BUFFER.

MOVE 0 TO CSVBUFOF
MOVE ' ' TO CSVBUF
MOVE 0 TO CONVLTH
MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
  CALL     CSVRSLT    USING FUNCCODE
                            T'NAME
                            L'NAME
                            D'NAME
```

Figure 25 Sample Program Using TOCSV (z/OS) (Page 1 of 3)

```
                            NAME
                            CSVBUF
                            CSVBUFSZ
                            CSVBUFOF
                            USRSYMBL
                            CONVLTH
                            CSVRETCD
                            CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'NAME = '   NAME
      HEXPRINT 'CSVBUF = ' CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF
;=================================================================
; CONVERT THE ACCOUNT FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE ACCOUNT FIELD IS PLACED AFTER THE NAME FIELD.
;=================================================================

; TO PLACE THE ACCOUNT FIELD AFTER THE NAME FIELD,
; ADD THE LENGTH OF THE CONVERTED NAME FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF



MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 0 TO CONVLTH
CALL CSVRSLT USING   FUNCCODE
                            T'ACCOUNT
                            L'ACCOUNT
                            D'ACCOUNT
                            ACCOUNT
                            CSVBUF
                            CSVBUFSZ
                            CSVBUFOF
                            USRSYMBL
                            CONVLTH
                            CSVRETCD
                            CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'ACCOUNT = ' ACCOUNT
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF
; ================================================================
; CONVERT THE BALANCE FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE BALANCE FIELD IS PLACED AFTER THE ACCOUNT FIELD.
; ================================================================

; TO PLACE THE BALANCE FIELD AFTER THE ACCOUNT FIELD,
; ADD THE LENGTH OF THE CONVERTED ACCOUNT FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF

MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 0 TO CONVLTH

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

CALL CSVRSLT USING   FUNCCODE
                            T'BALANCE
                            L'BALANCE
```

Figure 25 Sample Program Using TOCSV (z/OS) (Page 2 of 3)

```
                              D'BALANCE
                              BALANCE
                              CSVBUF
                              CSVBUFSZ
                              CSVBUFOF
                              USRSYMBL
                              CONVLTH
                              CSVRETCD
                              CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'BALANCE = ' BALANCE
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF


; ===================================================================
; REPORT THE CSV BUFFER
; ===================================================================
LIST CSVBUF

; ===================================================================
; SAVE THE CSV BUFFER FOR NEXT REQUEST IN ORDER TO USE
; CSVFROM FUNCTION
; ===================================================================

MOVE CSVBUF  TO CSVSTRING
CSVLEN = CONVLTH + CSVBUFOF
WRITE CSVFILE

ENDIF

ON FINAL
; ===================================================================
; TERMINATE THE ENVIRONMENT
; ===================================================================
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT USING FUNCCODE
```

Figure 25 Sample Program Using TOCSV (z/OS) (Page 3 of 3)

Output of Sample Program

```
                    CSV BUFFER
"MALTSBERGER, JOHN A       ","7099657","6.72"
"TURNER, HAROLD            ","7102194","5.12"
"SMITH, AUSTIN             ","7106246","5.12"
"HILL, MYRTLE G            ","9017828","11.20"
"HOPKINS, BARRY P          ","9000534","14.00"
"WORRELL, TED              ","9002626","13.00"
"RODEN,HAROLD              ","6215785","170.91"
"WHITE,ELMER               ","6208924","15.00"
"COX,WILLIAM               ","6211151","100.20"
"S FE EMP HOSP ASSN        ","7000227","36.00"
"SOOLEY,WILLIAM            ","6219004","15.00"
"ROUKE,CURTIS              ","6216412","48.00"
"WOODS, LOUISE             ","7012837",".00"
"FORBES, JOHN              ","7004753","11.20"
"RODRIGUEZ, EVARISTO       ","7012799","10.58"
"SMITH, JOHN               ","7010966","13.66"
"BLACK, LENORE             ","7039085","50.40"
"WHITE, ELMER              ","7011407","10.24"
"DE VAULT, JERRY           ","7033133","11.20"
"SIAS, EDWARDO             ","7023189",".00"
"REED,LOWELL D             ","2011719","127.22"
"HART,GRADDIE              ","2007177","128.37"
"BALDWIN,HARRY R           ","2006286","1108.92"
"CULLENDER,EVERRETT J      ","2013568","3886.33"
"JONES,RUFUS G             ","2009072","128.90"
"ZERING,WILLIAM G          ","2008831","124.95"
"NEBRENSKY,CARMEN C        ","6033873","3.40"
"CALLEROS,MARCELINO        ","6058523","5.05"
"SOARES,LAWERENCEW         ","6044751","8.15"
"TAYLOR,FLORENCE D         ","6060749","20.00"
"DIAZ,JOSE T               ","6061931","3.00"
"MC COY,FRANCES O          ","6034241","15.00"
"YOUNG,ADELE               ","6049672","15.00"
"FENZEE,WILLIAM            ","6031714","20.00"
"YEAGER,CLIFFORD M         ","6046819","15.04"
"MILLER,JOHN C             ","6105629","30.00"
"ZARATE,JOHN Q             ","6106501","101.20"
"GOULDING,JOHN             ","6099467","3.00"
"HEMPE,RALPH               ","6108415","4.25"
"ANGEL,JESUS               ","6098223","3.00"
"FLYNN,JOSEPH              ","6123031",".00"
"BROWN,STUART A            ","6113583","26.20"
"CONNER,AUBRY J            ","6202616","16.80"
```

Figure 26 Sample Output of TOCSV Function (z/OS)

Sample Program Using FROMCSV Function for z/OS

```
;=================================================================
; THIS PROGRAM USES THE CSV SYSTEM'S FROMCSV FUNCTION TO CONVERT CSV
; FORMATTED RECORDS TO ADVANTAGE VISION:RESULTS DATA FIELDS. THE
; PROGRAM READS THE CSV FILE, WHICH WAS CREATED IN THE SAMPLE PROGRAM
; USING THE TOCSV FUNCTION, AND CONVERTS THE CSV FORMATTED NAME,
; ACCOUNT, AND BALANCE VALUES TO VISION:RESULTS DATA FIELDS (NAME,
; ACCOUNT, AND BALANCE). THE DATA IN THE NAME, ACCOUNT, AND BALANCE
; FIELDS WILL BE PRINTED AS A REPORT.
; NOTE:  FOR DYLCOMRG TO PICK UP THE VALUE IN REGISTER 15, PLEASE SET
; RETCODE=Y IN THE DYLINSTL MACRO. FOR MORE INFORMATION, SEE THE SECTION
; "OPTIONAL PARAMETERS" IN THE VISION:RESULTS INSTALLATION GUIDE
;=================================================================


OPTION STRUCTURED2
FILE CSVFILE VB 120 LENGTH CSVLEN
CSVSTRING   120

WORKAREA
ACCOUNT  7   4 NU     (ACCOUNT'NUMBER)
NAME    25  85 CH     (CUSTOMER'NAME)
BALANCE  5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE 2 182 CH     (ACCOUNT'CODE)

WORKAREA
FUNCCODE   8 CH
CSVBUFSZ   4 BI
CSVBUFOF   4 BI
USRSYMBL   8 CH
CONVLTH    4 BI
CSVRETCD   4 BI
CSVRSNCD   4 BI
CSVBUF   120 CH (CSV BUFFER)

WORKAREA
 CHFIELD     2  CH
  REDEFINE CHFIELD
 BINFIELD    2  BI

REPORT1 120 WIDE

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

;=================================================================
; CONVERT THE CSV FORMATTED NAME VALUE AND PUT IT IN NAME.
; THE NAME VALUE IS LOCATED AT THE BEGINNING OF THE CSV BUFFER.
;=================================================================

MOVE 'FROMCSV' TO FUNCCODE
MOVE 0         TO CSVRETCD
MOVE 0         TO CSVRSNCD
MOVE CSVSTRING  TO CSVBUF
MOVE CSVLEN    TO CSVBUFSZ
MOVE 0         TO CSVBUFOF
MOVE 0         TO CONVLTH
MOVE BLANKS    TO NAME

CALL CSVRSLT USING   FUNCCODE
                     T'NAME
                     L'NAME
                     D'NAME
                     NAME
                     CSVBUF
                     CSVBUFSZ
                     CSVBUFOF
                     USRSYMBL
                     CONVLTH
                     CSVRETCD
                     CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
```

Figure 27 Sample Program Using FROMCSV (z/OS) (Page 1 of 3)

```
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'NAME = '   NAME
      HEXPRINT 'CSVBUF = ' CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF

;====================================================================
; CONVERT THE CSV FORMATTED ACCOUNT VALUE AND PUT IT IN ACCOUNT.
; THE ACCOUNT VALUE IS LOCATED AFTER THE NAME VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE ACCOUNT VALUE
; RESIDES IN THE BUFFER.
;====================================================================


MOVE 0      TO CSVRETCD
MOVE 0      TO CSVRSNCD
MOVE 0      TO CONVLTH
MOVE BLANKS TO ACCOUNT

CALL CSVRSLT USING   FUNCCODE
                     T'ACCOUNT
                     L'ACCOUNT
                     D'ACCOUNT
                     ACCOUNT
                     CSVBUF
                     CSVBUFSZ
                     CSVBUFOF
                     USRSYMBL
                     CONVLTH
                     CSVRETCD
                     CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'ACCOUNT = ' ACCOUNT
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF

; ====================================================================
; CONVERT THE CSV FORMATTED BALANCE VALUE AND PUT IT IN BALANCE.
; THE BALANCE VALUE IS PLACED AFTER THE ACCOUNT VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE BALANCE VALUE
; RESIDES IN THE BUFFER.
; ====================================================================


MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 0 TO CONVLTH
MOVE 0 TO BALANCE

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

CALL CSVRSLT USING   FUNCCODE
                     T'BALANCE
                     L'BALANCE
                     D'BALANCE
                     BALANCE
                     CSVBUF
                     CSVBUFSZ
                     CSVBUFOF
                     USRSYMBL
                     CONVLTH
```

Figure 27 Sample Program Using FROMCSV (z/OS) (Page 2 of 3)

```
                    CSVRETCD
                    CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'BALANCE = ' BALANCE
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF

; ====================================================================
; REPORT OF THE CSVFROM DATA FIELDS
; ====================================================================

LIST NAME ACCOUNT BALANCE

ON FINAL
; ====================================================================
; TERMINATE THE ENVIRONMENT
; ====================================================================
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT USING FUNCCODE
```

Figure 27 Sample Program Using FROMCSV (z/OS) (Page 3 of 3)

Output of Sample Program

```
      CUSTOMER                ACCOUNT       ACCOUNT
        NAME                  NUMBER        BALANCE
MALTSBERGER, JOHN A           7099657            6.72
TURNER, HAROLD                7102194            5.12
SMITH, AUSTIN                 7106246            5.12
HILL, MYRTLE G                9017828           11.20
HOPKINS, BARRY P              9000534           14.00
WORRELL, TED                  9002626           13.00
RODEN,HAROLD                  6215785          170.91
WHITE,ELMER                   6208924           15.00
COX,WILLIAM                   6211151          100.20
S FE EMP HOSP ASSN            7000227           36.00
SOOLEY,WILLIAM                6219004           15.00
ROUKE,CURTIS                  6216412           48.00
WOODS, LOUISE                 7012837             .00
FORBES, JOHN                  7004753           11.20
RODRIGUEZ, EVARISTO           7012799           10.58
SMITH, JOHN                   7010966           13.66
BLACK, LENORE                 7039085           50.40
WHITE, ELMER                  7011407           10.24
DE VAULT, JERRY               7033133           11.20
SIAS, EDWARDO                 7023189             .00
REED,LOWELL D                 2011719          127.22
HART,GRADDIE                  2007177          128.37
BALDWIN,HARRY R               2006286        1,108.92
CULLENDER,EVERRETT J          2013568        3,886.33
JONES,RUFUS G                 2009072          128.90
ZERING,WILLIAM G              2008831          124.95
NEBRENSKY,CARMEN C            6033873            3.40
CALLEROS,MARCELINO            6058523            5.05
SOARES,LAWERENCEW             6044751            8.15
TAYLOR,FLORENCE D             6060749           20.00
DIAZ,JOSE T                   6061931            3.00
MC COY,FRANCES O              6034241           15.00
YOUNG,ADELE                   6049672           15.00
FENZEE,WILLIAM                6031714           20.00
YEAGER,CLIFFORD M             6046819           15.04
MILLER,JOHN C                 6105629           30.00
ZARATE,JOHN Q                 6106501          101.20
GOULDING,JOHN                 6099467            3.00
HEMPE,RALPH                   6108415            4.25
ANGEL,JESUS                   6098223            3.00
FLYNN,JOSEPH                  6123031             .00
BROWN,STUART A                6113583           26.20
CONNER,AUBRY J                6202616           16.80
```

Figure 28 Sample Output of FROMCSV Function (z/OS)

Sample Program Using TOCSV Function for VSE

```
*****************************************************************
* THIS IS A SAMPLE APPLICATION THAT CALLS THE CSV TOOLKIT ROUTINE.
* NOTE: THE CALL STATEMENT CAN BE CODED IN EITHER OF THE
* FOLLOWING TWO FORMATS:
* 1) CALL CSVRSLT CDLOAD USING FUNCCODE, (ETC ...)
* 2) CALL CSVRSLT 90K USING FUNCCODE, (ETC ...)
*
;===============================================================
; THIS PROGRAM USES THE CSV SYSTEM'S TOCSV FUNCTION TO CREATE CSV
; FORMATTED RECORDS. THE PROGRAM READS THE ACCOUNTS RECEIVABLE SAMPLE
; FILE AND CONVERTS THE NAME, ACCOUNT, AND BALANCE FIELDS OF CERTAIN
```

Figure 29 Sample Program Using TOCSV (VSE) (Page 1 of 4)

```
; RECORDS TO CSV FORMATTED RECORDS. THE CSV FORMATTED RECORDS WILL BE
; PRINTED AS A REPORT AND WRITTEN TO A NEW VARIABLE SEQUENTIAL DATASET.
; THE VARIABLE SEQUENTIAL DATASET WILL BE USED AS INPUT BY THE SAMPLE
; PROGRAM USING THE FROMCSV FUNCTION.
; NOTE:  FOR DYLCOMRG TO PICK UP THE VALUE IN REGISTER 15, PLEASE SET
; RETCODE=Y IN THE DYLINSTL MACRO. FOR MORE INFORMATION, SEE THE SECTION
; "OPTIONAL PARAMETERS" IN THE VISION:RESULTS INSTALLATION GUIDE
;================================================================
OPTION STRUCTURED2
FILE ARFILE FB 352 5280 STATUS ARSTAT SYS014

ACCOUNT   7   4 NU     (ACCOUNT'NUMBER)
NAME     25  85 CH     (CUSTOMER'NAME)
BALANCE   5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE  2 182 CH     (ACCOUNT'CODE)

FILE CSVFILE VB 120 1244 OUTPUT FROM CSVFILE LENGTH CSVLEN SYS021
CSVSTRING   120


WORKAREA
FUNCCODE 8 CH

CSVBUFSZ 4 BI
CSVBUFOF 4 BI
USRSYMBL 8 CH
CONVLTH 4 BI
CSVRETCD 4 BI
CSVRSNCD 4 BI
CSVBUF   100 CH (CSV BUFFER)

WORKAREA
 CHFIELD     2  CH
  REDEFINE CHFIELD
 BINFIELD    2  BI

REPORT1 120 WIDE

ON ONE
MOVE 0TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 'TOCSV ' TO FUNCCODE
MOVE 120 TO CSVBUFSZ
MOVE 0 TO CSVBUFOF
MOVE 0 TO CONVLTH
ENDONE

IF (ACCTCODE EQ 'MA') AND (CSVRETCD EQ 0)

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

; ================================================================
; CONVERT THE NAME FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE NAME FIELD IS PLACED AT THE BEGINNING OF THE BUFFER.
; ================================================================

; SET THE OFFSET TO ZERO TO PLACE THE NAME FIELD
; AT THE BEGINNING OF THE BUFFER.
MOVE 0 TO CSVBUFOF
MOVE ' ' TO CSVBUF
MOVE 0 TO CONVLTH
MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
CALL CSVRSLT CDLOAD USING   FUNCCODE
T'NAME
L'NAME
D'NAME
NAME
CSVBUF
CSVBUFSZ
CSVBUFOF
USRSYMBL
CONVLTH
CSVRETCD
CSVRSNCD
```

Figure 29 Sample Program Using TOCSV (VSE) (Page 2 of 4)

```
MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'NAME = '   NAME
      HEXPRINT 'CSVBUF = ' CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF
 ; ================================================================
; CONVERT THE ACCOUNT FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE ACCOUNT FIELD IS PLACED AFTER THE NAME FIELD.
; ================================================================

; TO PLACE THE ACCOUNT FIELD AFTER THE NAME FIELD,
; ADD THE LENGTH OF THE CONVERTED NAME FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF

MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 0 TO CONVLTH



CALL CSVRSLT CDLOAD USING   FUNCCODE
T'ACCOUNT
L'ACCOUNT
D'ACCOUNT
ACCOUNT
CSVBUF
CSVBUFSZ
CSVBUFOF
USRSYMBL
CONVLTH
CSVRETCD
CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'ACCOUNT = ' ACCOUNT
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF
 ; ================================================================
; CONVERT THE BALANCE FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE BALANCE FIELD IS PLACED AFTER THE ACCOUNT FIELD.
; ================================================================

; TO PLACE THE BALANCE FIELD AFTER THE ACCOUNT FIELD,
; ADD THE LENGTH OF THE CONVERTED ACCOUNT FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF

MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 0 TO CONVLTH

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

CALL CSVRSLT CDLOAD USING   FUNCCODE
T'BALANCE
L'BALANCE
D'BALANCE
BALANCE
CSVBUF
CSVBUFSZ
CSVBUFOF
USRSYMBL
CONVLTH
CSVRETCD
```

Figure 29 Sample Program Using TOCSV (VSE) (Page 3 of 4)

```
CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'BALANCE = ' BALANCE
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF
 ; ===================================================================
; REPORT THE CSV BUFFER
; ===================================================================
LIST CSVBUF

; ===================================================================
; SAVE THE CSV BUFFER FOR NEXT REQUEST IN ORDER TO USE
; CSVFROM FUNCTION
; ===================================================================

MOVE CSVBUF  TO CSVSTRING
CSVLEN = CONVLTH + CSVBUFOF
WRITE CSVFILE


ENDIF

ON FINAL
; ===================================================================
; TERMINATE THE ENVIRONMENT
; ===================================================================
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT USING FUNCCODE
```

Figure 29 Sample Program Using TOCSV (VSE) (Page 4 of 4)

Output of Sample Program

```
                CSV BUFFER
"MALTSBERGER, JOHN A ","7099657","6.72"
"TURNER, HAROLD ","7102194","5.12"
"SMITH, AUSTIN ","7106246","5.12"
"HILL, MYRTLE G ","9017828","11.20"
"HOPKINS, BARRY P ","9000534","14.00"
"WORRELL, TED ","9002626","13.00"
"RODEN,HAROLD ","6215785","170.91"
"WHITE,ELMER ","6208924","15.00"
"COX,WILLIAM ","6211151","100.20"
"S FE EMP HOSP ASSN ","7000227","36.00"
"SOOLEY,WILLIAM ","6219004","15.00"
"ROUKE,CURTIS ","6216412","48.00"
"WOODS, LOUISE ","7012837",".00"
"FORBES, JOHN ","7004753","11.20"
"RODRIGUEZ, EVARISTO ","7012799","10.58"
"SMITH, JOHN ","7010966","13.66"
"BLACK, LENORE ","7039085","50.40"
"WHITE, ELMER ","7011407","10.24"
"DE VAULT, JERRY ","7033133","11.20"
"SIAS, EDWARDO ","7023189",".00"
"REED,LOWELL D ","2011719","127.22"
"HART,GRADDIE ","2007177","128.37"
"BALDWIN,HARRY R ","2006286","1108.92"
"CULLENDER,EVERRETT J ","2013568","3886.33"
"JONES,RUFUS G ","2009072","128.90"
"ZERING,WILLIAM G ","2008831","124.95"
"NEBRENSKY,CARMEN C ","6033873","3.40"
"CALLEROS,MARCELINO ","6058523","5.05"
"SOARES,LAWERENCEW ","6044751","8.15"
"TAYLOR,FLORENCE D ","6060749","20.00"
"DIAZ,JOSE T ","6061931","3.00"
"MC COY,FRANCES O ","6034241","15.00"
"YOUNG,ADELE ","6049672","15.00"
"FENZEE,WILLIAM ","6031714","20.00"
"YEAGER,CLIFFORD M ","6046819","15.04"
"MILLER,JOHN C ","6105629","30.00"
"ZARATE,JOHN Q ","6106501","101.20"
"GOULDING,JOHN ","6099467","3.00"
"HEMPE,RALPH ","6108415","4.25"
"ANGEL,JESUS ","6098223","3.00"
"FLYNN,JOSEPH ","6123031",".00"
"BROWN,STUART A ","6113583","26.20"
"CONNER,AUBRY J ","6202616","16.80"
```

Figure 30 Sample Output of TOCSV Function (VSE)

Sample Program Using FROMCSV Function for VSE

```
*********************************************************************
* THIS IS A SAMPLE APPLICATION THAT CALLS THE CSV TOOLKIT ROUTINE.
* NOTE: THE CALL STATEMENT CAN BE CODED IN EITHER OF THE
* FOLLOWING TWO FORMATS:
* 1) CALL CSVRSLT CDLOAD USING FUNCCODE, (ETC ...)
* 2) CALL CSVRSLT 90K USING FUNCCODE, (ETC ...)
*
;====================================================================
; THIS PROGRAM USES THE CSV SYSTEM'S FROMCSV FUNCTION TO CONVERT CSV
; FORMATTED RECORDS TO ADVANTAGE VISION:RESULTS DATA FIELDS. THE
; PROGRAM READS THE CSV FILE, WHICH WAS CREATED IN THE SAMPLE PROGRAM
; USING THE TOCSV FUNCTION, AND CONVERTS THE CSV FORMATTED NAME,
; ACCOUNT, AND BALANCE VALUES TO VISION:RESULTS DATA FIELDS (NAME,
; ACCOUNT, AND BALANCE). THE DATA IN THE NAME, ACCOUNT, AND BALANCE
; FIELDS WILL BE PRINTED AS A REPORT.
; NOTE:  FOR DYLCOMRG TO PICK UP THE VALUE IN REGISTER 15, PLEASE SET
; RETCODE=Y IN THE DYLINSTL MACRO. FOR MORE INFORMATION, SEE THE SECTION
; "OPTIONAL PARAMETERS" IN THE VISION:RESULTS INSTALLATION GUIDE
;====================================================================

OPTION STRUCTURED2
FILE CSVFILE VB 120 1244 LENGTH CSVLEN SYS014
CSVSTRING   120

WORKAREA
ACCOUNT  7   4 NU     (ACCOUNT'NUMBER)
NAME    25  85 CH     (CUSTOMER'NAME)
BALANCE  5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE 2 182 CH     (ACCOUNT'CODE)

WORKAREA
FUNCCODE   8 CH
CSVBUFSZ   4 BI
CSVBUFOF   4 BI
USRSYMBL   8 CH
CONVLTH    4 BI
CSVRETCD   4 BI
CSVRSNCD   4 BI
CSVBUF   120 CH (CSV BUFFER)

WORKAREA
 CHFIELD     2  CH
  REDEFINE CHFIELD
 BINFIELD    2  BI

REPORT1 120 WIDE

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

;====================================================================
; CONVERT THE CSV FORMATTED NAME VALUE AND PUT IT IN NAME.
; THE NAME VALUE IS LOCATED AT THE BEGINNING OF THE CSV BUFFER.
;====================================================================

MOVE 'FROMCSV' TO FUNCCODE
MOVE 0         TO CSVRETCD
MOVE 0         TO CSVRSNCD
MOVE CSVSTRING  TO CSVBUF
MOVE CSVLEN    TO CSVBUFSZ
MOVE 0         TO CSVBUFOF
MOVE 0         TO CONVLTH
MOVE BLANKS    TO NAME

CALL CSVRSLT CDLOAD USING   FUNCCODE
                            T'NAME
                            L'NAME
                            D'NAME
                            NAME
                            CSVBUF
                            CSVBUFSZ
```

Figure 31 Sample Program Using FROMCSV (VSE) (Page 1 of 3)

```
                              CSVBUFOF
                              USRSYMBL
                              CONVLTH
                              CSVRETCD
                              CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'NAME = '   NAME
      HEXPRINT 'CSVBUF = ' CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF

;*****************************************************************
; CONVERT THE CSV FORMATTED ACCOUNT VALUE AND PUT IT IN ACCOUNT.
; THE ACCOUNT VALUE IS LOCATED AFTER THE NAME VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE ACCOUNT VALUE
; RESIDES IN THE BUFFER.
;*****************************************************************


MOVE 0      TO CSVRETCD
MOVE 0      TO CSVRSNCD
MOVE 0      TO CONVLTH
MOVE BLANKS TO ACCOUNT

CALL CSVRSLT  CDLOAD USING   FUNCCODE
                             T'ACCOUNT
                             L'ACCOUNT
                             D'ACCOUNT
                             ACCOUNT
                             CSVBUF
                             CSVBUFSZ
                             CSVBUFOF
                             USRSYMBL
                             CONVLTH
                             CSVRETCD
                             CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'ACCOUNT = ' ACCOUNT
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF

; ***************************************************************
; CONVERT THE CSV FORMATTED BALANCE VALUE AND PUT IT IN BALANCE.
; THE BALANCE VALUE IS PLACED AFTER THE ACCOUNT VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE BALANCE VALUE
; RESIDES IN THE BUFFER.
; ***************************************************************


MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 0 TO CONVLTH
MOVE 0 TO BALANCE

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

CALL CSVRSLT  CDLOAD USING   FUNCCODE
                             T'BALANCE
                             L'BALANCE
```

Figure 31 Sample Program Using FROMCSV (VSE) (Page 2 of 3)

```
                                D'BALANCE
                                BALANCE
                                CSVBUF
                                CSVBUFSZ
                                CSVBUFOF
                                USRSYMBL
                                CONVLTH
                                CSVRETCD
                                CSVRSNCD

MOVE DYLCOMRG TO CHFIELD
   IF BINFIELD NE 0
      HEXPRINT 'REGISTER 15 = '     BINFIELD
               'CSV RETURN CODE = ' CSVRETCD
               'CSV REASON CODE = ' CSVRSNCD
      HEXPRINT 'BALANCE = ' BALANCE
      HEXPRINT 'CSVBUF = '  CSVBUF
      MOVE 77 TO DYLRETURN
      STOP
   ENDIF

; ===================================================================
; REPORT OF THE CSVFROM DATA FIELDS
; ===================================================================

LIST NAME ACCOUNT BALANCE

ON FINAL
; ===================================================================
; TERMINATE THE ENVIRONMENT
; ===================================================================
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT CDLOAD USING FUNCCODE
```

Figure 31 Sample Program Using FROMCSV (VSE) (Page 3 of 3)

Output of Sample Program

```
       CUSTOMER                ACCOUNT       ACCOUNT
         NAME                  NUMBER        BALANCE
  MALTSBERGER, JOHN A          7099657            6.72
  TURNER, HAROLD               7102194            5.12
  SMITH, AUSTIN                7106246            5.12
  HILL, MYRTLE G               9017828           11.20
  HOPKINS, BARRY P             9000534           14.00
  WORRELL, TED                 9002626           13.00
  RODEN,HAROLD                 6215785          170.91
  WHITE,ELMER                  6208924           15.00
  COX,WILLIAM                  6211151          100.20
  S FE EMP HOSP ASSN           7000227           36.00
  SOOLEY,WILLIAM               6219004           15.00
  ROUKE,CURTIS                 6216412           48.00
  WOODS, LOUISE                7012837             .00
  FORBES, JOHN                 7004753           11.20
  RODRIGUEZ, EVARISTO          7012799           10.58
  SMITH, JOHN                  7010966           13.66
  BLACK, LENORE                7039085           50.40
  WHITE, ELMER                 7011407           10.24
  DE VAULT, JERRY              7033133           11.20
  SIAS, EDWARDO                7023189             .00
  REED,LOWELL D                2011719          127.22
  HART,GRADDIE                 2007177          128.37
  BALDWIN,HARRY R              2006286        1,108.92
  CULLENDER,EVERRETT J         2013568        3,886.33
  JONES,RUFUS G                2009072          128.90
  ZERING,WILLIAM G             2008831          124.95
  NEBRENSKY,CARMEN C           6033873            3.40
  CALLEROS,MARCELINO           6058523            5.05
  SOARES,LAWERENCEW            6044751            8.15
  TAYLOR,FLORENCE D            6060749           20.00
  DIAZ,JOSE T                  6061931            3.00
```

Figure 32 Sample Output of FROMCSV Function (VSE) (Page 1 of 2)

```
MC COY,FRANCES O              6034241           15.00
YOUNG,ADELE                   6049672           15.00
FENZEE,WILLIAM                6031714           20.00
YEAGER,CLIFFORD M             6046819           15.04
MILLER,JOHN C                 6105629           30.00
ZARATE,JOHN Q                 6106501          101.20
GOULDING,JOHN                 6099467            3.00
HEMPE,RALPH                   6108415            4.25
ANGEL,JESUS                   6098223            3.00
FLYNN,JOSEPH                  6123031             .00
BROWN,STUART A                6113583           26.20
CONNER,AUBRY J                6202616           16.80
```

Figure 32 Sample Output of FROMCSV Function (VSE) (Page 2 of 2)

Copy Support Routines

A call to the CSVRSLT API is required for each field that you add to your CSV file. This requirement for many fields could result in a program that is long and verbose. However, the use of CSV copy support routines, CSVWORK and CSVCALL, make CSV conversion easier and the output more concise.

CSVWORK and CSVCALL are copy members included in the VISION:Results installation source library. CSVWORK contains VISION:Results field definitions for most of the required CSV parameters. CSVCALL contains code that accepts one parameter (the field value), then generates complete code for a call to CSVRSLT.

Sample Program of CSVWORK Code

Note: #L is the length for CSV buffer or CSVBUF. The default length is 100. To change the length to 120, for example, code COPY CSVWORK #L=120.

```
$DEFAULT #L=100
$DEND
 WORKAREA
  FUNCCODE   8  CH
  CSVBUFSZ    4  BI
  CSVBUFOF    4  BI
  USRSYMBL   8  CH
  CONVLTH     4  BI
  CSVRETCD   4  BI
  CSVRSNCD   4  BI
  CSVBUF   #L   CH

WORKAREA
 CHFIELD     2  CH
  REDEFINE CHFIELD
 BINFIELD    2  BI
```

Figure 33 Sample of CSVWORK Code

Sample Program of CSVCALL Code

Note: #N is the field or data name that is being converted. [CDLOAD] is provided for VSE only. For DYLCOMRG to pick up the value in Register 15, set RETCODE=Y in the DYLINSTL macro. For more information, see the *Advantage VISION:Results for z/OS Installation Guide*.

```
MOVE 0 TO CONVLTH
MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
CALL CSVRSLT  [CDLOAD] USING
FUNCCODE T'#N  L'#N  D'#N  #N
CSVBUF  CSVBUFSZ  CSVBUFOF
USRSYMBL  CONVLTH  CSVRETCD  CSVRSNCD

         MOVE DYLCOMRG TO CHFIELD
IF BINFIELD NE 0
     HEXPRINT 'REGISTER 15 = '               BINFIELD
                    'CSV RETURN CODE = '  CSVRETCD
                        'CSV REASON CODE = '  CSVRSNCD
     HEXPRINT '#N = '  #N
     HEXPRINT 'CSVBUF = '  CSVBUF
     MOVE 77 TO DYLRETURN
     STOP
ENDIF
```

Figure 34 Sample of CSVCALL Code

Differences in Required Code for CSVRSLT

The following tables illustrate the difference in the required code for the use of CSVRSLT.

Coding Required without CSVWORK:	Coding Required with CSVWORK:
WORKAREA FUNCCODE 8 CH CSVBUFSZ 4 BI CSVBUFOF 4 BI USRSYMBL 8 CH CONVLTH 4 BI CSVRETCD 4 BI CSVRSNCD 4 BI CSVBUF 100 CH WORKAREA CHFIELD 2 CH REDEFINE CHFIELD BINFIELD 2 BI	COPY CSVWORK

Coding Required without CSVCALL:	Coding Required with CSVCALL:
MOVE 0 TO CONVLTH MOVE 0 TO CSVRETCD MOVE 0 TO CSVRSNCD CALL CSVRSLT [CDLOAD] USING FUNCCODE T'NAME L'NAME D'NAME NAME CSVBUF CSVBUFSZ CSVBUFOF USRSYMBL CONVLTH CSVRETCD CSVRSNCD MOVE DYLCOMRG TO CHFIELD IF BINFIELD NE 0 HEXPRINT 'REGISTER 15 = ' BINFIELD 'CSV RETURN CODE = ' CSVRETCD 'CSV REASON CODE = ' CSVRSNCD HEXPRINT 'NAME = ' NAME HEXPRINT 'CSVBUF = ' CSVBUF MOVE 77 TO DYLRETURN STOP ENDIF	COPY CSVCALL #N=NAME

Note: [CDLOAD] for VSE only. For DYLCOMRG to pick up the value in Register 15, set RETCODE=Y in the DYLINSTL macro.

Sample Programs Using TOCSV Function with CSVWORK and CSVCALL

This section provides sample TOCSV programs using copy CSVWORK and CSVCALL members.

Sample Program of CSVWORK and CSVCALL Code (z/OS)

```
;=================================================================
; THIS PROGRAM USES THE CSV SYSTEM'S TOCSV FUNCTION TO CREATE CSV
; FORMATTED RECORDS. THE PROGRAM READS THE ACCOUNTS RECEIVABLE SAMPLE
; FILE AND CONVERTS THE NAME, ACCOUNT, AND BALANCE FIELDS OF CERTAIN
; RECORDS TO CSV FORMATTED RECORDS. THE CSV FORMATTED RECORDS WILL BE
; PRINTED AS A REPORT AND WRITTEN TO A NEW VARIABLE SEQUENTIAL DATASET.
; THE VARIABLE SEQUENTIAL DATASET WILL BE USED AS INPUT BY THE SAMPLE
; PROGRAM USING THE FROMCSV FUNCTION.
;=================================================================

OPTION STRUCTURED2
FILE ARFILE FB 352  STATUS ARSTAT
ACCOUNT  7   4 NU     (ACCOUNT'NUMBER)
NAME    25  85 CH     (CUSTOMER'NAME)
BALANCE  5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE 2 182 CH     (ACCOUNT'CODE)
```

Figure 35 Sample Program Using TOCSV Function with CSVWORK and CSVCALL (z/OS) (Page 1 of 3)

```
FILE CSVFILE VB 120 OUTPUT FROM CSVFILE LENGTH CSVLEN
CSVSTRING   120

COPY CSVWORK

REPORT1 120 WIDE

ON ONE
MOVE 0 TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 'TOCSV ' TO FUNCCODE
MOVE 120 TO CSVBUFSZ
MOVE 0 TO CSVBUFOF
MOVE 0 TO CONVLTH
ENDONE

IF (ACCTCODE EQ 'MA') AND (CSVRETCD EQ 0)

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

;=================================================================
; CONVERT THE NAME FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE NAME FIELD IS PLACED AT THE BEGINNING OF THE BUFFER.
;=================================================================

; SET THE OFFSET TO ZERO TO PLACE THE NAME FIELD
; AT THE BEGINNING OF THE BUFFER.

MOVE 0 TO CSVBUFOF
MOVE ' ' TO CSVBUF
COPY CSVCALL #N=NAME

;=================================================================
; CONVERT THE ACCOUNT FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE ACCOUNT FIELD IS PLACED AFTER THE NAME FIELD.
;=================================================================

; TO PLACE THE ACCOUNT FIELD AFTER THE NAME FIELD,
; ADD THE LENGTH OF THE CONVERTED NAME FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF

COPY CSVCALL #N=ACCOUNT

; ================================================================
; CONVERT THE BALANCE FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE BALANCE FIELD IS PLACED AFTER THE ACCOUNT FIELD.
; ================================================================

; TO PLACE THE BALANCE FIELD AFTER THE ACCOUNT FIELD,
; ADD THE LENGTH OF THE CONVERTED ACCOUNT FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

COPY CSVCALL #N=BALANCE

; ================================================================
; REPORT THE CSV BUFFER
; ================================================================
LIST CSVBUF (CSV BUFFER)

; ================================================================
; SAVE THE CSV BUFFER FOR NEXT REQUEST IN ORDER TO USE
; CSVFROM FUNCTION
; ================================================================

MOVE CSVBUF  TO CSVSTRING
CSVLEN = CONVLTH + CSVBUFOF
WRITE CSVFILE
```

Figure 35 Sample Program Using TOCSV Function with CSVWORK and CSVCALL (z/OS) (Page 2 of 3)

```
ENDIF

ON FINAL
; ================================================================
; TERMINATE THE ENVIRONMENT
; ================================================================
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT USING FUNCCODE
```

Figure 35 Sample Program Using TOCSV Function with CSVWORK and CSVCALL (z/OS) (Page 3 of 3)

Sample Program of CSVWORK and CSVCALL Code (VSE)

```
*****************************************************************
* THIS IS A SAMPLE APPLICATION THAT CALLS THE CSV TOOLKIT ROUTINE.
* NOTE: THE CALL STATEMENT CAN BE CODED IN EITHER OF THE
* FOLLOWING TWO FORMATS:
* 1) CALL CSVRSLT CDLOAD USING FUNCCODE, (ETC ...)
* 2) CALL CSVRSLT 90K USING FUNCCODE, (ETC ...)
*
;================================================================
; THIS PROGRAM USES THE CSV SYSTEM'S TOCSV FUNCTION TO CREATE CSV
; FORMATTED RECORDS. THE PROGRAM READS THE ACCOUNTS RECEIVABLE SAMPLE
; FILE AND CONVERTS THE NAME, ACCOUNT, AND BALANCE FIELDS OF CERTAIN
; RECORDS TO CSV FORMATTED RECORDS. THE CSV FORMATTED RECORDS WILL BE
; PRINTED AS A REPORT AND WRITTEN TO A NEW VARIABLE SEQUENTIAL DATASET.
; THE VARIABLE SEQUENTIAL DATASET WILL BE USED AS INPUT BY THE SAMPLE
; PROGRAM USING THE FROMCSV FUNCTION.
;================================================================
OPTION STRUCTURED2
FILE ARFILE FB 352 5280 STATUS ARSTAT SYS014
ACCOUNT   7   4 NU     (ACCOUNT'NUMBER)
NAME     25  85 CH     (CUSTOMER'NAME)
BALANCE   5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE  2 182 CH     (ACCOUNT'CODE)

FILE CSVFILE VB 120 1244 OUTPUT FROM CSVFILE LENGTH CSVLEN SYS021
CSVSTRING   120


COPY CSVWORK

REPORT1 120 WIDE

ON ONE
MOVE 0TO CSVRETCD
MOVE 0 TO CSVRSNCD
MOVE 'TOCSV ' TO FUNCCODE
MOVE 120 TO CSVBUFSZ
MOVE 0 TO CSVBUFOF
MOVE 0 TO CONVLTH
ENDONE

IF (ACCTCODE EQ 'MA') AND (CSVRETCD EQ 0)

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

; ================================================================
; CONVERT THE NAME FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE NAME FIELD IS PLACED AT THE BEGINNING OF THE BUFFER.
; ================================================================

; SET THE OFFSET TO ZERO TO PLACE THE NAME FIELD
; AT THE BEGINNING OF THE BUFFER.
MOVE 0 TO CSVBUFOF
MOVE ' ' TO CSVBUF
```

Figure 36 Sample Program Using TOCSV Function with CSVWORK and CSVCALL (VSE) (Page 1 of 2)

```
COPY CSVCALL #N=NAME

; ===================================================================
; CONVERT THE ACCOUNT FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE ACCOUNT FIELD IS PLACED AFTER THE NAME FIELD.
; ===================================================================

; TO PLACE THE ACCOUNT FIELD AFTER THE NAME FIELD,
; ADD THE LENGTH OF THE CONVERTED NAME FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF

COPY CSVCALL #N=ACCOUNT


; ===================================================================
; CONVERT THE BALANCE FIELD TO CSV FORMAT AND PUT IT IN THE BUFFER.
; THE BALANCE FIELD IS PLACED AFTER THE ACCOUNT FIELD.
; ===================================================================

; TO PLACE THE BALANCE FIELD AFTER THE ACCOUNT FIELD,
; ADD THE LENGTH OF THE CONVERTED ACCOUNT FIELD TO THE OFFSET.
CSVBUFOF = CONVLTH + CSVBUFOF

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

COPY CSVCALL #N=BALANCE

; ===================================================================
; REPORT THE CSV BUFFER
; ===================================================================
LIST CSVBUF (CSV BUFFER)


; ===================================================================
; SAVE THE CSV BUFFER FOR NEXT REQUEST IN ORDER TO USE
; CSVFROM FUNCTION
; ===================================================================

MOVE CSVBUF  TO CSVSTRING
CSVLEN = CONVLTH + CSVBUFOF
WRITE CSVFILE


ENDIF

ON FINAL
; ===================================================================
; TERMINATE THE ENVIRONMENT
; ===================================================================
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT USING FUNCCODE
```

Figure 36 Sample Program Using TOCSV Function with CSVWORK and CSVCALL (VSE) (Page 2 of 2)

Sample Programs Using FROMCSV Function with CSVWORK and CSVCALL

This section provides sample FROMCSV programs using copy CSVWORK and CSVCALL members.

Sample Program of CSVWORK and CSVCALL Code (z/OS)

```
;=================================================================
; THIS PROGRAM USES THE CSV SYSTEM'S FROMCSV FUNCTION TO CONVERT CSV
; FORMATTED RECORDS TO ADVANTAGE VISION:RESULTS DATA FIELDS. THE
; PROGRAM READS THE CSV FILE, WHICH WAS CREATED IN THE SAMPLE PROGRAM
; USING THE TOCSV FUNCTION, AND CONVERTS THE CSV FORMATTED NAME,
; ACCOUNT, AND BALANCE VALUES TO VISION:RESULTS DATA FIELDS (NAME,
; ACCOUNT, AND BALANCE). THE DATA IN THE NAME, ACCOUNT, AND BALANCE
; FIELDS WILL BE PRINTED AS A REPORT.
;=================================================================


OPTION STRUCTURED2
FILE CSVFILE VB 120 LENGTH CSVLEN
CSVSTRING   120

WORKAREA
ACCOUNT  7   4 NU     (ACCOUNT'NUMBER)
NAME    25  85 CH     (CUSTOMER'NAME)
BALANCE  5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE 2 182 CH     (ACCOUNT'CODE)

COPY CSVWORK #L=120

REPORT1 120 WIDE

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

;=================================================================
; CONVERT THE CSV FORMATTED NAME VALUE AND PUT IT IN NAME.
; THE NAME VALUE IS LOCATED AT THE BEGINNING OF THE CSV BUFFER.
;=================================================================

MOVE 'FROMCSV' TO FUNCCODE
MOVE CSVSTRING  TO CSVBUF
MOVE CSVLEN     TO CSVBUFSZ
MOVE 0          TO CSVBUFOF
MOVE BLANKS     TO NAME

COPY CSVCALL #N=NAME

;=================================================================
; CONVERT THE CSV FORMATTED ACCOUNT VALUE AND PUT IT IN ACCOUNT.
; THE ACCOUNT VALUE IS LOCATED AFTER THE NAME VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE ACCOUNT VALUE
; RESIDES IN THE BUFFER.
;=================================================================

COPY CSVCALL #N=ACCOUNT

; ================================================================
; CONVERT THE CSV FORMATTED BALANCE VALUE AND PUT IT IN BALANCE.
; THE BALANCE VALUE IS PLACED AFTER THE ACCOUNT VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE BALANCE VALUE
; RESIDES IN THE BUFFER.
; ================================================================
```

Figure 37 Sample Program Using FROMCSV Function with CSVWORK and CSVCALL (z/OS) (Page 1 of 2)

```
MOVE 0 TO BALANCE

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

COPY CSVCALL #N=BALANCE

; ===================================================================
; REPORT OF THE CSVFROM DATA FIELDS
; ===================================================================

LIST NAME ACCOUNT BALANCE

ON FINAL
; ===================================================================
; TERMINATE THE ENVIRONMENT
; ===================================================================
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT USING FUNCCODE
```

Figure 37 Sample Program Using FROMCSV Function with CSVWORK and CSVCALL (z/OS) (Page 2 of 2)

Sample Program of CSVWORK and CSVCALL Code (VSE)

```
*********************************************************************
* THIS IS A SAMPLE APPLICATION THAT CALLS THE CSV TOOLKIT ROUTINE.
* NOTE: THE CALL STATEMENT CAN BE CODED IN EITHER OF THE
* FOLLOWING TWO FORMATS:
* 1) CALL CSVRSLT CDLOAD USING FUNCCODE, (ETC ...)
* 2) CALL CSVRSLT 90K USING FUNCCODE, (ETC ...)
*
;===================================================================
; THIS PROGRAM USES THE CSV SYSTEM'S FROMCSV FUNCTION TO CONVERT CSV
; FORMATTED RECORDS TO ADVANTAGE VISION:RESULTS DATA FIELDS. THE
; PROGRAM READS THE CSV FILE, WHICH WAS CREATED IN THE SAMPLE PROGRAM
; USING THE TOCSV FUNCTION, AND CONVERTS THE CSV FORMATTED NAME,
; ACCOUNT, AND BALANCE VALUES TO VISION:RESULTS DATA FIELDS (NAME,
; ACCOUNT, AND BALANCE). THE DATA IN THE NAME, ACCOUNT, AND BALANCE
; FIELDS WILL BE PRINTED AS A REPORT.
;===================================================================


OPTION STRUCTURED2
FILE CSVFILE VB 120 1244 LENGTH CSVLEN SYS014
CSVSTRING   120

WORKAREA
ACCOUNT  7   4 NU     (ACCOUNT'NUMBER)
NAME    25  85 CH     (CUSTOMER'NAME)
BALANCE  5 170 PD 2 A (ACCOUNT'BALANCE)
ACCTCODE 2 182 CH     (ACCOUNT'CODE)

COPY CSVWORK #L=120

REPORT1 120 WIDE

; SET UP THE USER DESIGNATED CHARACTERS
MOVE '",\. - ' TO USRSYMBL

;===================================================================
; CONVERT THE CSV FORMATTED NAME VALUE AND PUT IT IN NAME.
; THE NAME VALUE IS LOCATED AT THE BEGINNING OF THE CSV BUFFER.
;===================================================================

MOVE 'FROMCSV' TO FUNCCODE
MOVE CSVSTRING  TO CSVBUF
MOVE CSVLEN    TO CSVBUFSZ
```

Figure 38 Sample Program Using FROMCSV Function with CSVWORK and CSVCALL (VSE) (Page 1 of 2)

```
MOVE 0          TO CSVBUFOF
MOVE BLANKS     TO NAME

COPY CSVCALL #N=NAME

;*****************************************************************
; CONVERT THE CSV FORMATTED ACCOUNT VALUE AND PUT IT IN ACCOUNT.
; THE ACCOUNT VALUE IS LOCATED AFTER THE NAME VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE ACCOUNT VALUE
; RESIDES IN THE BUFFER.
;*****************************************************************


MOVE BLANKS TO ACCOUNT

COPY CSVCALL #N=ACCOUNT

; *****************************************************************
; CONVERT THE CSV FORMATTED BALANCE VALUE AND PUT IT IN BALANCE.
; THE BALANCE VALUE IS PLACED AFTER THE ACCOUNT VALUE.
; NOTE: CSVRSLT ROUTINE AUTOMATICALLY SETS THE CSV BUFFER
; OFFSET PARAMETER, CSVBUFOF, TO POINT TO WHERE THE BALANCE VALUE
; RESIDES IN THE BUFFER.
; *****************************************************************


MOVE 0 TO BALANCE

; CHANGE THE COMMA TO A BLANK FOR THE LAST FIELD IN THE BUFFER.
MOVE '" \. - ' TO USRSYMBL

COPY CSVCALL #N=BALANCE

; *****************************************************************
; REPORT OF THE CSVFROM DATA FIELDS
; *****************************************************************

LIST NAME ACCOUNT BALANCE

ON FINAL
; *****************************************************************
; TERMINATE THE ENVIRONMENT
; *****************************************************************
MOVE 'TERM ' TO FUNCCODE
CALL CSVRSLT CDLOAD USING FUNCCODE
```

Figure 38 Sample Program Using FROMCSV Function with CSVWORK and CSVCALL (VSE) (Page 2 of 2)

Chapter 6: PCFILE and PCWRITE

The PCFILE statement and the PCWRITE command write data to a sequential data set or a temporary JES for download by VISION:Intraaccess, or to the VISION:Journey for Windows VSAM download file. This feature is only available to z/OS VISION:Results users who also subscribe to VISION:Intraaccess or VISION:Journey for Windows.

- When used with VISION:Intraaccess, the data is written to either a temporary JES file, or a variable length cataloged data set. VISION:Intraaccess then reads the data, transforms it, and downloads it to the client PC.
- When used with VISION:Journey for Windows, the data is written to a temporary sequential file in the VSAM download format and, in a subsequent VISION:Results step, the sequential file is transferred to the VISION:Journey for Windows VSAM download file. The temporary sequential file is a variable length file.

PCFILE Statement

The PCFILE statement that defines the PC file is formatted as follows:

```
PCFILE fileid
FIELDS !dataname!
[MAXBLKSIZE nnnnn]
[STATUS dataname]
```

For the PCFILE statement to function successfully, the following conditions must be met:

- The fileid must not appear in any FILE statement. JCL statements for the fileid must be included in the VISION:Results job stream or be processed by the ALLOCATE command.
- The fields specified in the FIELDS clause of the PCFILE statement must have been previously defined in the program.
- The LRECL of the temporary sequential file must be at least equal to the sum of the length of all the fields specified in the FIELDS clause plus 24 bytes. The LRECL cannot exceed 8024 bytes.
- The MAXBLKSIZE must be at least as large as the LRECL; otherwise, the temporary sequential file is unblocked.

- If the MAXBLKSIZE clause is omitted, the temporary sequential file is unblocked. For z/OS only, if the MAXBLKSIZE clause is omitted but a block size is defined in the JCL for the file, that block size is used.
- The STATUS data name is a self-defining, one-byte field. When the PCWRITE command is successfully executed, the STATUS data name contains the value Y; otherwise the STATUS data name contains the value N.
- No more than 255 PCFILE statements can be defined in one program request.
- A PCFILE statement cannot be placed after an ON CHANGE IN, ON FINAL, ON END OF INPUT, or ON END OF SORTING statement.

PCWRITE Command

When the PCWRITE command is executed, a record is written to the temporary sequential file created by the PCFILE statement.

The PCWRITE command is formatted as follows:

```
PCWRITE fileid
```

Sample Program

Following is a sample program using the PCFILE statement and the PCWRITE command.

```
OPTION STRUCTURED2

FILE SYSIN CARDS
     INKEY        5   1
     INCODE       1
     INACCT       6
     INBALANCE    7

WORKAREA
     WKCODE       5

PCFILE LOTUSFIL
     STATUS PCSTATUS   MAXBLKSIZE 2000
     FIELDS INKEY INACCT INBALANCE WKCODE

IF INCODE EQ 'A'
     MOVE 'ACE' TO WKCODE
     PCWRITE LOTUSFIL
     IF PCSTATUS NE 'Y'
        PRINT 'ACE WRITE FAILURE'
        STOP
     ENDIF
ENDIF

IF INCODE EQ 'H'
```

Figure 39 Example Program Using PCFILE and PCWRITE (Page 1 of 2)

```
     MOVE 'HEART' TO WKCODE
     PCWRITE LOTUSFIL
     IF PCSTATUS NE 'Y'
        PRINT 'HEART WRITE FAILURE'
        STOP
     ENDIF
ENDIF

REJECT
```

Figure 39 Example Program Using PCFILE and PCWRITE (Page 2 of 2)

Chapter 7: Library Functions

TIMEGET (Getting the Current Time)

The TIMEGET program is an external routine that can be called by VISION:Results (in a z/OS or CMS environment) to obtain the current time of day. The format of the call is:

```
CALL TIMEGET USING Parm1 Parm2 Parm3 [Parm4]
```

Parm1 Required 2-byte field for hours.

Parm2 Required 2-byte field for minutes.

Parm3 Required 2-byte field for seconds.

Parm4 Optional 1-byte format request. Valid requests are C for zoned decimal or P for packed decimal. If the parameter is omitted, P is assumed.

```
WORKAREA
   THEHOUR       2 PD     ;CURRENT HOUR
   THEMINUTE     2 PD     ;CURRENT MINUTE
   THESECOND     2 PD     ;CURRENT SECOND
   FMT           1 CH VALUE 'P'

CALL TIMEGET USING THEHOUR THEMINUTE THESECOND FMT
```

Figure 40 TIMEGET Program

CVROMAN (Converting to Roman Numerals)

The CVROMAN program is an external routine that converts binary to Roman numerals. It can be called from a VISION:Results program with the following parameters:

4	BI input:	The value to be converted.
2	BI output:	Length of the converted value.
20	CH output:	Value after conversion.
2	BI output:	Return code (0=OK, 1=error).

```
WORKAREA
     ANUMBER      4  BI     ;NUMBER TO BE CONVERTED
     THESIZE      2  BI     ;SIZE OF OUTPUT AREA USED
     CONVERTED    20 CH     ;CONVERTED VALUE
     RETCODE      2  BI     ;RETURN CODE (0=Ok, 1=error)

MOVE 2344 TO ANUMBER
CALL CVROMAN USING ANUMBER THESIZE CONVERTED RETCODE
IF (RETCODE EQ 0)
   you now have the converted date
ELSE
   there is a problem
ENDIF
```

Figure 41 CVROMAN Program

Limitations:

- No negative numbers (error code is set if a negative number is used).
- No numbers greater than 10,000 (error code is set if a number greater than 10,000 is used).

FROMROM (Converting from Roman Numerals)

The FROMROM program is an external routine that can be called from VISION:Results. It is the inverse of the CVROMAN program; it takes Roman numerals and converts them to binary.

It is called using the following parameters:

30	CH	input:	The Roman numeral to convert.
2	BI	input:	Length of the Roman numeral.
2	BI	input:	The converted value.
2	BI	output:	A return code (0=ok).

```
WORKAREA
   INCOMING     30  CH     ;VALUE TO BE CONVERTED
   THESIZE       2  BI     ;LENGTH OF INCOMING
   OUTGOING      2  BI     ;CONVERTED VALUE
   RETCODE       2  BI     ;

MOVE "MCMLXXXIX" TO INCOMING
MOVE 9 TO THESIZE
CALL FROMROM USING INCOMING THESIZE OUTGOING RETCODE
IF (RETCODE EQ 0)
   you now have the converted date in OUTGOING
ELSE
   there is a problem
ENDIF
```

Figure 42 FROMROM Program

A non-zero return code is generated if any letter other than those used in Roman numerals (M=thousand, D=five hundred, C=hundred, L=fifty, X=ten, V=five, and I=one) is encountered in the input.

Standard notation is required. For example, “XXXX” is invalid; “XL” is the only accepted format for the number 40. Similar rules apply for 4, 9, 90, 400, and 900.

CVSTATE (Converting from State Abbreviations)

The CVSTATE program is an external routine that can be called from a VISION:Results program to convert a two-letter state abbreviation into the full state name. The parameters are:

2	CH input:	The state abbreviation.
20	CH output:	The actual state name.
2	BI output:	Return code (0=OK).

```
WORKAREA
   ABBREV     2  CH     ;STATE ABBREVIATION
   STATENAME  20 CH     ;STATE NAME
   RETCODE    2  BI     ;RETURN CODE (0=Ok)

MOVE “CA” TO ABBREV
CALL CVSTATE USING ABBREV STATENAME RETCODE
IF (RETCODE EQ 0)
   the requested abbreviation for “CALIFORNIA” was found
ELSE
   the abbreviation is not understood
ENDIF
```

Figure 43 CVSTATE Program

CVWORDS (Converting from Numbers to English)

The CVWORDS program is an external routine that can be called from a VISION:Results program to convert a number into English. The following parameters are used:

Parm1	A 4-byte binary number to be converted.
Parm2	On input, this is the length of the Parm3 work area. The length is from 20 bytes through 256 bytes. On output, this is the length of the text being returned. If the length is zero on return, an error has occurred and the first 20 bytes to the buffer has a message (such as, BUFFER OVERFLOW).
Parm3	Work area address. Be sure this matches the length in Parm2 and is adequate for the largest number to be converted.

```
WORKAREA
   ANUMBER   4 BI     ;NUMBER TO CONVERT
   THESIZE   2 BI     ;SIZE AFTER CONVERSION
   ENGLISH   120 CH   ;ENGLISH TRANSLATION

MOVE 120 TO THESIZE
MOVE 155 TO ANUMBER
CALL CVWORDS USING ANUMBER THESIZE ENGLISH
This leaves "ONE HUNDRED FIFTY-FIVE" in the work area variable named "ENGLISH" and the
value 22 in "THESIZE."
```

Figure 44 CVWORDS Program

Negative numbers are supported; the output has the qualifier “MINUS” in front.

DYLDAYWK (Calculating the Day of the Week)

DYLDAYWK is a VISION:Results subroutine that calculates the day of the week for any input date between January 1, 1700 and December 31, 2400. DYLDAYWK accepts a date in one of seven formats. DYLDAYWK automatically adjusts for leap years.

DYLDAYWK can be used on both z/OS and VSE systems; it is self-relocating, serially reusable, and less than 2K long.

Coding Requirements

DYLDAYWK is invoked by using a VISION:Results CALL statement. The CALL statement requires three parameters to be passed that identify the input date, input format type, and the output day/error message areas.

```
CALL DYLDAYWK 2K USING Parm1 Parm2 Parm3
```

Entry	Description
DYLDAYWK	Name of called subroutine.
2K	Memory size for DYLDAYWK (VSE only).
(parameters)	Designate areas (parameters) to contain input date, the format type, and the day of the week or error messages.

Entry	Description
Parm1	Defines a 10-byte character input area of numeric month, day, and year. Format can be one of 7 types: MMDDYYYY (type 1) DDMMYYYY (type 2) YYYYMMDD (type 3) MM/DD/YYYY (type 4) DD/MM/YYYY (type 5) YYYY/MM/DD (type 6) YYYYDDD (type 7 – Julian)
Parm2	Defines a 1-byte character field containing a value from 1 to 7 corresponding to the format of the date as above.
Parm3	Identifies a 10-byte area to hold the day of the week or any error messages returned from DYLDAYWK.

Date fields must contain numeric data (except for slashes between month, day, and year on types 4, 5, 6). Months must be numeric values between 01 and 12. Years must be numeric values between 1700 and 2400. Julian days must be between 01 and 365 (366 for leap years).

Error Messages

If an error is detected while executing DYLDAYWK, a diagnostic error message is placed in the Parm3 area. If no error is detected, the day of the week is returned.

Message	Cause
ADJ ERROR	Leap-year calculation error within DYLDAYWK itself. Be sure all input fields are in the proper format for the type specified, and are numeric. Contact CA Technical Support for assistance. See Contacting CA Technical Support on page 10.
CALC ERROR	Calculation error within DYLDAYWK itself. Be sure all input fields are in the proper format for the type specified, and are numeric. Contact CA Technical Support for assistance. See Contacting CA Technical Support on page 10.
DATE ERROR	The input date is not numeric (for types 4, 5, and 6, the MM, DD, and YYYY fields must be numeric and there must be slashes between these fields). Be sure the date is numeric, the type specified is correct for that format of the date, and there is data in the input record.

Message	Cause
DAY ERROR	For formats 1 through 6: A value less than 01 or greater than the correct number of days for that month was detected. Be sure that the input is numeric and DD in the date is not greater than the correct number of days for that month, particularly if the month is February. For format 7: A value of less than 01 or greater than 365 (366 for leap year) was detected. Be sure the date is numeric and between those limit values.
MONTH ERR	A value that is not between 01 and 12 was detected in the input date for formats 1 through 6. Be sure the input is numeric and not less than 01 or greater than 12.
TYPE ERROR	The input type is not 1 through 7. Be sure the type specified is between these limits and is correct for the date format entered.
YEAR ERROR	Input year is less than 1700 or greater than 2400. Be sure the date is numeric and YYYY value is between these limits.

DYLDAYWK Example

In this example, nine input dates are supplied between January 1, 1700 and December 31, 2400 from instream data. The following VISION:Results statements accomplish the task.

Input

```
FILE SYSIN CARD
     DATE 10
     TYPE  1

WORKAREA
     DAY  10    VALUE ' '  REINIT

CALL DYLDAYWK  USING DATE TYPE DAY

LIST DATE TYPE DAY

T1  'CALCULATE THE DAY OF THE WEEK'
T2  'DYLDAYWK EXAMPLE' WITH 2 AFTER

FIN
```

Figure 45 DYLDAYWK Input Example

Output

```
CALCULATE THE DAY OF THE WEEK
          DYLDAYWK EXAMPLE

   DATE         TYPE         DAY

01591218         3        YEAR ERROR
24/01/1815       5        TUESDAY
04071940         2        THURSDAY
1800/05/25       6        SUNDAY
11111811         A        TYPE ERROR
08/20/2000       4        SUNDAY
1700001          7        FRIDAY
01171953         1        SATURDAY
17431129         3        FRIDAY
```

Figure 46 DYLDAYWK Output Example

DYLADAYS (Calculating Future and Past Dates)

DYLADAYS is a VISION:Results subroutine that calculates a date by adding or subtracting a number of days from a specific date. The input date can be specified in seven different formats, six in Gregorian and one in Julian. The output date is in the same format as the input date. The input date can be increased or decreased by 1 to 3,500,000 days.

DYLADAYS can be used on both z/OS and VSE systems; it is serially reusable, self-relocating, and less than 3K long.

Coding Requirements

DYLADAYS is called by using a VISION:Results CALL statement. The CALL statement requires four parameters that identify the date format, input date, number of days to add or subtract, and output date area.

```
CALL DYLADAYS 3K USING PARM1 PARM2 PARM3 PARM4
```

Entry	Description
DYLADAYS	Name of called subroutine.
3K	Memory size for DYLADAYS (VSE only).
(parameters)	Designate areas (parameters) to contain date format, input date, number of days to add or subtract, and output date area.

Entry	Description
Parm1	Defines a 1-byte input date format. MMDDYYYY (type 1) DDMMYYYY (type 2) YYYYMMDD (type 3) MM/DD/YYYY (type 4) DD/MM/YYYY (type 5) YYYY/MM/DD (type 6) YYYYDDD (type 7)
Parm2	Identifies a 7-10 byte field depending on the date format that holds the input date.
Parm3	Identifies a 4-byte packed field that holds the number of days to add or subtract from the input data. If positive, the number of days is added. If negative, the number of days is subtracted.
Parm4	The same format and size as the Parm2 input date field. It is also an invalid message output field. The date format must be a numeric value between 1 and 7. Month must be a numeric value between 01 and 12. Maximum days added or subtracted is 3,500,000 days. Any invalid input date is detected by DYLADAYS and the message INVALID is placed in the Parm4 field.

Error Messages

If an error is detected while executing DYLADAYS, the word INVALID is placed in the Parm4 field. The possible causes are:

Invalid code	Must be a numeric value between 1 and 7.
Invalid month	Must be a numeric value between 01 and 12.
Invalid day	Must be numeric and not larger than the number of days within its related month. If the input date is in Julian format, 366 is the maximum days for a leap year and 365 is the maximum days for a non-leap year.
Invalid year	Must be a numeric value between 400 and 9999.

DYLADAYS Example

In this example, you are shown how to write a VISION:Results program that uses the DYLADAYS subroutine to calculate the date you want. The input data is taken from instream data.

Input

```
FILE SYSIN CARD
     CODEIN      1
     DATEIN     10
     DAYS        8
     REDEFINE DAYS
        SIGN     1
        DAY      7  NU
     DATEOUT    10

WORKAREA
     CODE        1
     INDATE     10
     DAYPD       4  PD
     OUTDATE    10

MOVE CODEIN TO CODE
MOVE DATEIN TO INDATE
MOVE DAY    TO DAYPD
IF SIGN EQ '-'
     DAYPD = DAYPD * -1
ENDIF
MOVE DATEOUT TO OUTDATE

CALL DYLADAYS  USING CODE INDATE DAYPD OUTDATE

LIST CODE INDATE SIGN DAY OUTDATE

T1  'CALCULATE WANTED DATE'
T2  'DYLADAYS EXAMPLE' WITH 2 AFTER
FIN
```

Figure 47 DYLADAYS Example

Output

```
CALCULATE DATE  WANTED
                      DYLADAYS EXAMPLE
CODE       INDATE       SIGN       DAY        OUTDATE

 4       02/28/1996      +       0000001     02/29/1996
 5       28/02/2000      +       0000366     28/02/2001
 6       1996/02/29      -       0000366     1995/02/28
```

Figure 48 DYLADAYS Example

CONVDATE (Calculating and Converting Dates)

CONVDATE is a VISION:Results callable routine that is invoked to perform various date calculations and conversions. Using CONVDATE, you can convert dates from one format to another (two-character year to four-character year) and adjust a date forward or backward.

The CONVDATE routine requires a date table containing the information necessary to perform the different date conversion functions. A default table named CONVDATT is provided. You can customize this table, or generate a new table using the CONVDATT source member provided. If the table name is changed, then function 12 must be invoked before other functions to load the renamed table.

CONVDATT

The user table, CONVDATT, contains three areas of data. The first area defines the status of each of the seven days of the week. The second area is a collection of control information. The third area is a holiday dates table.

Customization of the user date table is done by creating and executing an Assembler program that invokes the WEEKDAY and the HOLIDAY macros.

The WEEKDAY macro must be invoked for each day of the week, beginning with day 1, which is Monday, and continuing consecutively (day 2, day 3,...). The macro consists of two keyword parameters, DAY and DAYLEN. You must supply the day name, abbreviated to 3 characters (MON, TUE, WED,...,SUN). A default value (WHOLE) is supplied for the DAYLEN parameter. For a normal work day, you do not need to specify this parameter. The acceptable values you can specify are HALF (indicating the normal length of the day is half day) or OFF (indicating that no work is done on that day).

```
WEEKDAY DAY=MON | TUE | WED | THU | FRI | SAT | SUN
        [,DAYLEN=HALF | OFF | WHOLE]
```

After the WEEKDAY macro has been specified as necessary, you begin creating entries for your holiday dates table. This is accomplished with the HOLIDAY macro. The macro contains five keyword symbolic parameters: DATE, DAYLEN, WKDAEND, CHAR2, and CHAR4. The syntax is:

```
HOLIDAY DATE=MMDDYYYY,
         DAYLEN=WHOLE | HALF
         [WKDAEND=1|2|3|4|5|6|7]
         [CHAR2=19|20]
         [CHAR4=19|20]
```

The last three parameters are used only the first time the macro is invoked. If not specified at the time of the first invocation of the macro, default values will be taken. These parameters, and information summed from the DAYLEN parameter of the WEEKDAY macro, are used to create the control information that has three items. They are the normal work week length, the day that designates the end of the work week, and the two-part binary flags used to indicate century year prefix digits for two- and four-character date default values.

The normal week length is obtained by summing the DAYLEN parameters of the WEEKDAY macro. (Note: The DAYLEN parameters in the WEEKDAY macro and the HOLIDAY macro are unrelated.) The WKDAEND (default = 7) is set to a value of 1 through 7 designating Monday, Tuesday,... as the day known in your installation as the end of the work week.

The CHAR2 and CHAR4 parameters are assigned values of 19 or 20. These values are the prefixes that are used when expanding a date to the four-character year format. CHAR4 is only used in Function 01 to format the output date. CHAR2 is used in functions 02, 03, 04, 05, 06, 07, 08, 09, 10, and 11 to format the input dates.

When creating the holiday dates, remember the dates must be entered in ascending sequence. Dates are entered in the format MMDDYYYY where MM represents the month (01=January,...), DD represents the day (01,...), and YYYY is the four-character year. Your dates can regress as far back as 01011900 (January 1, 1900) and project as far ahead as 12312099 (December 31, 2099).

Within the restrictions noted, you can indicate any combination of dates as holidays. Values that you can use for the DAYLEN parameter are WHOLE (default) or HALF. It is possible to have multiple tables. To do so requires each table to be link edited into your load libraries under a different name. Function 12 (explained later) of CONVDATT can be used to switch back and forth between tables.

Note: The JCL and source are provided in the Installation Source Library (see the *Advantage VISION:Results for z/OS Installation Guide*).

The following sample of JCL shows how to create a new CONVDATT table. The macro values shown in this sample are the ones used to create the CONVDATT table delivered with Version 6.0.

```
//* YOUR JOB CARD GOES HERE
//ASMLINK PROC DISK=SYSDA,
//             LOADLIB='YOUR.RESULTS.LOAD.LIBRARY',          OLD
//             NEWNAM='NEW.DATE.TABLE.NAME',                 NEW
//             SOUT='*',
//             SOURCLB='YOUR.RESULTS.SOURCE.LIBRARY'         OLD
//*
//*********************************************************************
//*
//HLASM   EXEC PGM=ASMA90,REGION=2M,TIME=(,40),                        X
//         PARM=(OBJECT,NODECK,'LINECOUNT(48)','USING(MAP,WARN(3))',   X
//              TERM,'XREF(FULL)')
//SYSLIB       DD  DSN=&SOURCLB,DISP=SHR
//             DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSLIN   DD  DSNAME=&&OBJ,UNIT=&DISK,SPACE=(3040,(80,80),,,ROUND),
//             DISP=(MOD,PASS),
//             DCB=(BLKSIZE=3040,LRECL=80,RECFM=FBS,BUFNO=1)
//SYSUT1   DD  DSNAME=&SYSUT1,UNIT=&DISK,SPACE=(1700,(400,50)),
//             SEP=(SYSLIN)
//SYSUT2   DD  DSNAME=&SYSUT2,UNIT=&DISK,SPACE=(1700,(400,50)),
//             SEP=(SYSLIN,SYSUT1)
//SYSUT3   DD  DSNAME=&SYSUT3,UNIT=&DISK,SPACE=(1700,(400,50)),
//             SEP=(SYSLIN,SYSUT1,SYSUT2)
//SYSPRINT DD  SYSOUT=&SOUT
//SYSPUNCH DD  DUMMY
//SYSTERM  DD  SYSOUT=&SOUT
//*
//STEP02  EXEC PGM=IEWL,REGION=2M,PARM='XREF,LIST,NCAL',
//             COND=(0,NE)
//* LINK EDIT THE DATE TABLE.
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUT1   DD  UNIT=&DISK,SPACE=(1024,(250,20))
//SYSLMOD  DD  DSN=&LOADLIB.(&NEWNAM),DISP=SHR
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//*
//          PEND
/*
//STEP1   EXEC ASMLINK
//SYSIN   DD  *
*    HOLIDAY AND DATE TABLE
*====> BE SURE TO EXAMINE/UPDATE DATES FOR YOUR INSTALLATION,
*====> AND THE CHAR2 AND CHAR4 PARAMETERS IF YOU WANT TO CHANGE
*====>   THE DEFAULT CENTURY PREFIXES.
         WEEKDAY DAY=MON
         WEEKDAY DAY=TUE
         WEEKDAY DAY=WED
```

Figure 49 CONVDATT JCL and Table for z/OS (Page 1 of 3)

```
         WEEKDAY DAY=THU
         WEEKDAY DAY=FRI
         WEEKDAY DAY=SAT,DAYLEN=HALF
         WEEKDAY DAY=SUN,DAYLEN=OFF
         HOLIDAY DATE=01022006,DAYLEN=WHOLE,CHAR2=20,CHAR4=20
*                                                NEW YEARS, 2006
         HOLIDAY DATE=01162006,DAYLEN=WHOLE     MARTIN LUTHER KING DAY
         HOLIDAY DATE=02202006,DAYLEN=WHOLE     PRESIDENT'S DAY
         HOLIDAY DATE=05292006,DAYLEN=WHOLE     MEMORIAL DAY
         HOLIDAY DATE=07042006,DAYLEN=WHOLE     INDEPENDENCE DAY
         HOLIDAY DATE=09042006,DAYLEN=WHOLE     LABOR DAY
         HOLIDAY DATE=10092006,DAYLEN=WHOLE     COLUMBUS DAY
         HOLIDAY DATE=11102006,DAYLEN=WHOLE     VETERANS DAY
         HOLIDAY DATE=11232006,DAYLEN=WHOLE     THANKSGIVING DAY
         HOLIDAY DATE=12242006,DAYLEN=HALF      CHRISTMAS EVE
         HOLIDAY DATE=12252006,DAYLEN=WHOLE     CHRISTMAS
         HOLIDAY DATE=12292006,DAYLEN=HALF      NEW YEAR'S EVE
*
         HOLIDAY DATE=01012007,DAYLEN=WHOLE     NEW YEARS, 2007
         HOLIDAY DATE=01152007,DAYLEN=WHOLE     MARTIN LUTHER KING DAY
         HOLIDAY DATE=02192007,DAYLEN=WHOLE     PRESIDENT'S DAY
         HOLIDAY DATE=05282007,DAYLEN=WHOLE     MEMORIAL DAY
         HOLIDAY DATE=07042007,DAYLEN=WHOLE     INDEPENDENCE DAY
         HOLIDAY DATE=09032007,DAYLEN=WHOLE     LABOR DAY
         HOLIDAY DATE=10082007,DAYLEN=WHOLE     COLUMBUS DAY
         HOLIDAY DATE=11122007,DAYLEN=WHOLE     VETERANS DAY
         HOLIDAY DATE=11222007,DAYLEN=WHOLE     THANKSGIVING DAY
         HOLIDAY DATE=12242007,DAYLEN=HALF      CHRISTMAS EVE
         HOLIDAY DATE=12252007,DAYLEN=WHOLE     CHRISTMAS
         HOLIDAY DATE=12312007,DAYLEN=HALF      NEW YEAR'S EVE
*
         HOLIDAY DATE=01012008,DAYLEN=WHOLE     NEW YEARS, 2008
         HOLIDAY DATE=01212008,DAYLEN=WHOLE     MARTIN LUTHER KING DAY
         HOLIDAY DATE=02182008,DAYLEN=WHOLE     PRESIDENT'S DAY
         HOLIDAY DATE=05262008,DAYLEN=WHOLE     MEMORIAL DAY
         HOLIDAY DATE=07042008,DAYLEN=WHOLE     INDEPENDENCE DAY
         HOLIDAY DATE=09012008,DAYLEN=WHOLE     LABOR DAY
         HOLIDAY DATE=10132008,DAYLEN=WHOLE     COLUMBUS DAY
         HOLIDAY DATE=11112008,DAYLEN=WHOLE     VETERANS DAY
         HOLIDAY DATE=11272008,DAYLEN=WHOLE     THANKSGIVING DAY
         HOLIDAY DATE=12242008,DAYLEN=HALF      CHRISTMAS EVE
         HOLIDAY DATE=12252008,DAYLEN=WHOLE     CHRISTMAS
         HOLIDAY DATE=12312008,DAYLEN=HALF      NEW YEAR'S EVE
*
         HOLIDAY DATE=01012009,DAYLEN=WHOLE     NEW YEARS, 2009
         HOLIDAY DATE=01192009,DAYLEN=WHOLE     MARTIN LUTHER KING DAY
         HOLIDAY DATE=02162009,DAYLEN=WHOLE     PRESIDENT'S DAY
         HOLIDAY DATE=05252009,DAYLEN=WHOLE     MEMORIAL DAY
         HOLIDAY DATE=07032009,DAYLEN=WHOLE     INDEPENDENCE DAY
         HOLIDAY DATE=09072009,DAYLEN=WHOLE     LABOR DAY
         HOLIDAY DATE=10122009,DAYLEN=WHOLE     COLUMBUS DAY
         HOLIDAY DATE=11112009,DAYLEN=WHOLE     VETERANS DAY
         HOLIDAY DATE=11262009,DAYLEN=WHOLE     THANKSGIVING DAY
         HOLIDAY DATE=12252009,DAYLEN=WHOLE     CHRISTMAS
         HOLIDAY DATE=12312009,DAYLEN=HALF      NEW YEAR'S EVE
*
         HOLIDAY DATE=01012010,DAYLEN=WHOLE     NEW YEARS, 2010
         HOLIDAY DATE=01182010,DAYLEN=WHOLE     MARTIN LUTHER KING DAY
         HOLIDAY DATE=02152010,DAYLEN=WHOLE     PRESIDENT'S DAY
         HOLIDAY DATE=05312010,DAYLEN=WHOLE     MEMORIAL DAY
         HOLIDAY DATE=07052010,DAYLEN=WHOLE     INDEPENDENCE DAY
         HOLIDAY DATE=09062010,DAYLEN=WHOLE     LABOR DAY
         HOLIDAY DATE=10112010,DAYLEN=WHOLE     COLUMBUS DAY
         HOLIDAY DATE=11112010,DAYLEN=WHOLE     VETERANS DAY
         HOLIDAY DATE=11252010,DAYLEN=WHOLE     THANKSGIVING DAY
         HOLIDAY DATE=12242010,DAYLEN=WHOLE     CHRISTMAS
         HOLIDAY DATE=12312010,DAYLEN=HALF      NEW YEAR'S EVE
*
         HOLIDAY DATE=01032011,DAYLEN=WHOLE     NEW YEARS, 2011
         HOLIDAY DATE=01172011,DAYLEN=WHOLE     MARTIN LUTHER KING DAY
         HOLIDAY DATE=02212011,DAYLEN=WHOLE     PRESIDENT'S DAY
```

Figure 49 CONVDATT JCL and Table for z/OS (Page 2 of 3)

```
HOLIDAY DATE=05302011,DAYLEN=WHOLE     MEMORIAL DAY
HOLIDAY DATE=07042011,DAYLEN=WHOLE     INDEPENDENCE DAY
HOLIDAY DATE=09052011,DAYLEN=WHOLE     LABOR DAY
HOLIDAY DATE=10102011,DAYLEN=WHOLE     COLUMBUS DAY
HOLIDAY DATE=11112011,DAYLEN=WHOLE     VETERANS DAY
HOLIDAY DATE=11242011,DAYLEN=WHOLE     THANKSGIVING DAY
HOLIDAY DATE=12262011,DAYLEN=WHOLE     CHRISTMAS
HOLIDAY DATE=12302011,DAYLEN=HALF      NEW YEAR'S EVE
END
```

Figure 49 CONVDATT JCL and Table for z/OS (Page 3 of 3)

Printing of the Date Tables

The utility program CONVTPRT is provided to print the contents of the date tables.

The report produced by CONVTPRT is in two parts. The first part prints the weekday and miscellaneous parts of the table. This details the defined length of each day, the day designated as the end of the week, the normal work week length, and default prefixes for two-character year format date values. The remainder of the report is a listing of the entries in the holiday dates portion of the table.

Sample of JCL to print the dates table

```
//*  PRINT OUT DATE TABLE
//*     STEP1:  USE DEFAULT CONVDATT
//*     STEP2:  SPECIFY, USING PARM, WHICH DATE TABLE
//STEP1  EXEC PGM=CONVTPRT
//STEPLIB  DD   DISP=SHR,DSN='your.Results.loadlib'
//SYSPRINT DD   SYSOUT=*
//*
//STEP2  EXEC PGM=CONVTPRT,PARM=MYDATTT
//STEPLIB  DD   DISP=SHR,DSN='your.Results.loadlib'
//SYSPRINT DD   SYSOUT=*
```

CONVDATE

There are 14 different functions performed by CONVDATE; they all use the following general syntax:

```
CALL CONVDATE [6K] USING FUNCTION PARM1 PARM2 PARM3 PARM4 PARM5 PARM6
```

The parameters are described in the following table. Not all parameters apply for every function that CONVDATE performs. There are specific function descriptions following the table to help you determine which parameters to use for a given function.

Entry	Description
CONVDATE	Name of the called routine.
[6K]	VSE only. This is the size of the called routine. Alternatively, you can specify CDLOAD.

Entry	Description
FUNCTION	The name of a 2-character data name that contains the function to be performed. Allowed values are 01 through 13.
Parm1 – Parm6	Lists data names (parameters) which contain the input date to convert or calculate, the input and output date format return areas, and optional parameters as described below.
Input Date	A data name containing the date upon which CONVDATE is to convert or base calculations. For any of the Date Formats listed below except Lilian (LLLL), this entry must be a character field, with a format and length matching its corresponding FORMAT field (see Date Format on page 188).

Date Format	A data name that contains the date format, which must be a character field with one of the following values:
	MMDDYY DDMMYY YYMMDD
	DDMMYYYY YYDDD MM/DD/YY
	DD/MM/YY YY/MM/DD DD/MM/YYYY
	MMDDYYYY YYYYDDD YYYYMMDD
	MM/DD/YYYY YYYY/MM/DD LLLL
	where: MM is the month
	DD is the day
	YY is the two-character year
	YYYY is the four-character year
	DDD is the Julian day of the year
	LLLL is the number of elapsed days since October 15, 1582 (Lilian date). The input must be a 4-byte binary field.
	Note: This format is valid only for function codes 01 and 02.
	`FUNCTION 2 VALUE '01'`
	`DATEIN 10 VALUE '960115'`
	`DATEFORM 10 VALUE 'MMDDYY '`
	`DATEOUT 10`
	`CALL CONVDATE USING FUNCTION DATEIN DATEFORM DATEOUT`
	Note: At least 1 blank must be present at the end of the DATE format field.

Result Field	Required only by certain functions, the result field contains the result of a date calculation or conversion. The size of this area varies by function. See the appropriate function description for details.

Entry	Description
Number of Adjusting Days	When using the function to adjust a date (function 8), this area contains the number of days you want to add or subtract to and from the DATE. This factor must be a five-character number if positive (add to DATE) or a six-character number with a leading minus sign (subtract from DATE). See the example for Function 08: Adjust a Date Up or Down on page 196.
Last Day of Week Code	When using the function to compute a week ending date for a given date, this area contains the day code (1-7) of the last day of the week. The default code is 7 (Sunday).
Prefixes for Four-Character Years	An optional operand for producing a four-character year. If this optional parameter is not specified, the value assigned is taken from the values contained in the date table. Functions 1 through 5 and 7 through 10 can have values of: 4, 19, 20. (These operands do not apply to Function 6.) Function 11 has one of two values: 19, 20 where the values are defined as follows: 4 four-character year input format or use the year prefix default from the user date table 19 two-character year prefix to expand an input date to output as a four-character year 20 two-character year prefix to expand an input date to output as a four-character year

CONVDATE Functions

Function 01: Convert Julian Date to Gregorian/Lilian Date

```
CALL CONVDATE  USING  FUNCTION  DATE  DATE_FORMAT RETURN_AREA
[YEAR_PFX]
```

The RETURN_AREA parameter must be defined as either a 10-byte character field containing a GREGORIAN date (MM/DD/YYYY), or a 4-byte binary field containing a LILIAN date (for example, 143254). When the YEAR_PFX parameter is used and it specifies a value of 4, the input DATE parameter must be a 7-byte character field with a 4-digit year Julian date (YYYYDDD). Otherwise, the input DATE parameter must be a 5-byte character field with a 2-digit year Julian date (YYDDD). The DATE_FORMAT parameter determines whether a LILIAN or GREGORAN output date is returned.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '01'
DATEIN       10   VALUE '96015'
DATEFORM     10   VALUE 'MMDDYY '
DATEOUT      10


CALL CONVDATE USING FUNCTION DATEIN DATEFORM DATEOUT
LIST DATEIN (JULIAN DATE) DATEFORM (OUTPUT DATE'FORMAT)
     DATEOUT (GREGORIAN DATE)

T1
T1+50  'CONVERT JULIAN TO GREGORIAN'

STOP
```

Figure 50 Convert Julian Date to Gregorian Date Input

Output

```
                CONVERT JULIAN TO GREGORIAN
  JULIAN DATE     OUTPUT DATE     GREGORIAN DATE
                    FORMAT
       96015          MMDDYY          011596
```

Figure 51 Convert Julian Date to Gregorian Date Output

Input

```
FILE ININ DUMMY
WORKAREA
FUNCTION   2  VALUE  '01'
DATEIN   10  VALUE  '75001'
DATEFORM  11  VALUE  'LLLL'
DATEOUT    4  BI
CALL CONVDATE USING FUNCTION DATEIN DATEFORM DATEOUT
LIST DATEIN  DATEFORM DATEOUT
STOP
```

Figure 52 Convert Julian Date to Lilian Date Input

Output

```
DATEIN        DATEFORM   DATEOUT
75001          LLLL      143254
```

Figure 53 Convert Julian Date to Lilian Date Output

Function 02: Convert Gregorian/Lilian Date to Julian Date

```
CALL CONVDATE   USING   FUNCTION     DATE    DATE_FORMAT
RETURN_AREA  [YEAR_PFX]
```

The RETURN_AREA parameter returns a 5-byte character field with a 2-digit year Julian date (YYDDD). When the YEAR_PFX parameter is used and it specifies a value of 4, then the RETURN_AREA parameter returns a 7-byte character field with a 4-digit year Julian date (YYYYDDD). The DATE_FORMAT parameter contains the format of the LILIAN or GREGORAN date type to be converted.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '02'
DATEIN       10   VALUE '01151996'
DATEFORM     10   VALUE 'MMDDYYYY '
DATEOUT      10
DATEPFX       2   VALUE '4'

CALL CONVDATE USING FUNCTION DATEIN DATEFORM DATEOUT
LIST DATEIN (GREGORIAN DATE) DATEFORM (INPUT DATE'FORMAT)
     DATEOUT (JULIAN DATE)

CALL CONVDATE USING FUNCTION DATEIN DATEFORM DATEOUT DATEPFX
LIST DATEIN (GREGORIAN DATE) DATEFORM (INPUT DATE'FORMAT)
     DATEOUT (JULIAN DATE)
T1
T1+50       'CONVERT GREGORIAN TO JULIAN'

STOP
```

Figure 54 Convert Gregorian Date to Julian Date Input

Output

```
                CONVERT GREGORIAN TO JULIAN
GREGORIAN DATE     INPUT DATE     JULIAN DATE
                    FORMAT
   01151996        MMDDYYYY         96015
   01151996        MMDDYYYY       1996015
```

Figure 55 Convert Gregorian Date to Julian Date Output

Input

```
FILE ININ DUMMY
WORKAREA
FUNCTION    2  VALUE  '02'
DATEIN      4  BI VALUE 152423
DATEFORM1  10 VALUE 'LLLL '
DATEFORM2   7  VALUE  'YYYYDDD'
DATEOUT     7
YRPFX4      2  VALUE   '4 '

CALL CONVDATE USING FUNCTION DATEIN DATEFORM1 DATEOUT YRPFX4

LIST DATEIN(LILIAN INPUT) DATEFORM1(LILIAN DATE'FORMAT)
     DATEFORM2(OUTPUT DATE'FORMAT) DATEOUT(JULIAN DATE)
STOP
```

Figure 56 Convert Lillian Date to Gregorian Date Input

Output

```
LILIAN INPUT     LILIAN DATE     OUTPUT DATE     JULIAN DATE
                   FORMAT          FORMAT
 0000152423      LLLL            YYYYDDD          2000039
```

Figure 57 Convert Lillian Date to Gregorian Date Output

Function 03: Compute Number of Days Between Two Dates

```
CALL CONVDATE   USING    FUNCTION  DATE1 DATE1_FORMAT
DATE2 DATE2_FORMAT RETURN_AREA [YEAR_PFX]
```

The RETURN_AREA data name must be defined as a 3-byte packed decimal field.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '03'
DATE1IN      10   VALUE '96/01/15 '
DAT1FORM     10   VALUE 'YY/MM/DD '
DATE2IN      10   VALUE '96/01/25'
DAT2FORM     10   VALUE 'YY/MM/DD '
NUMBDAYS     03   PD           ;RETURN_AREA must be defined as 03 PD.
CALL CONVDATE USING FUNCTION DATE1IN DAT1FORM DATE2IN DAT2FORM NUMBDAYS

LIST DATE1IN(FIRST DATE) DAT1FORM (FIRST DATE'FORMAT)
 DATE2IN(SECOND DATE) DAT2FORM (SECOND DATE'FORMAT)
 NUMBDAYS(NUMBER OF DAYS'BETWEEN DATES)

T1
T1+50       'COMPUTE #DAYS BET 2 DATES '

STOP
```

Figure 58 Number of Days Between Two Dates Input

Output

```
                          COMPUTE #DAYS BET 2 DATES
FIRST DATE     FIRST DATE     SECOND DATE     SECOND DATE     NUMBER OF DAYS
                 FORMAT                         FORMAT         BETWEEN DATES
 96/01/15       YY/MM/DD       96/01/25        YY/MM/DD           00010
```

Figure 59 Number of Days Between Two Dates Output

Function 04: Convert a Date to a Day of the Week

```
CALL CONVDATE   USING   FUNCTION   DATE_IN DATE_FORMAT
RETURN_AREA      [YEAR_PFX]
```

The RETURN_AREA data name must be defined as a 10-byte character field. The first byte is the day number, a digit from 1 to 7 (1=mon, 2=tues, and so forth), followed by the complete day name (MONDAY, TUESDAY, and so forth).

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '04'
DATEIN       10   VALUE '02251996'
DATEFORM     10   VALUE 'MMDDYYYY '
WEEKDAY      10
 REDEFINE AT WEEKDAY
 WEEK_NUM  1
 WEEK_DAY  9

CALL CONVDATE USING FUNCTION DATEIN DATEFORM WEEKDAY
LIST DATEIN (INPUT DATE) DATEFORM (INPUT DATE'FORMAT)
     WEEK_NUM (DAY NUMBER) WEEK_DAY (DAY OF'WEEK)

T1
T1+50       'CONVERT DATE TO DAY OF WEEK'

STOP
```

Figure 60 Convert a Date to a Day of the Week Input

Output

```
                CONVERT DATE TO DAY OF WEEK
INPUT DATE      INPUT DATE      DAY NUMBER  DAY OF
                  FORMAT                      WEEK
02251996         MMDDYYYY           7       SUNDAY
```

Figure 61 Convert a Date to a Day of the Week Output

Function 05: Compute Month Ending Date for a Given Date

```
CALL CONVDATE USING FUNCTION DATE_IN FORMAT_IN DATE_RETURN FORMAT_RETURN [YEAR_PFX]
```

The RETURN_AREA data name must be defined as a 10-byte character field.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '05'
DATEIN       10   VALUE '96/01/15'
DAT1FORM     10   VALUE 'YY/MM/DD '
DATEOUT      10
DAT2FORM     10   VALUE 'YYDDD '


CALL CONVDATE USING FUNCTION DATEIN DAT1FORM DATEOUT DAT2FORM
LIST DATEIN (INPUT DATE) DAT1FORM (INPUT DATE'FORMAT)
 DATEOUT (MONTH ENDING'DATE) DAT2FORM(OUTPUT DATE'FORMAT)

T1
T1+50       'COMPUTE MONTH ENDING DATE'

STOP
```

Figure 62 Compute Month Ending Date for a Given Date Input

Output

```
                        COMPUTE MONTH ENDING DATE
   INPUT DATE      INPUT DATE      MONTH ENDING      OUTPUT DATE
                     FORMAT            DATE            FORMAT
    96/01/15       YY/MM/DD          96031            YYDDD
```

Figure 63 Compute Month Ending Date for a Given Date Output

Function 06: Compute Number of Days Left in a Calendar Month

```
CALL CONVDATE   USING   FUNCTION     DATE_IN     DATE_FORMAT
     RETURN_AREA
```

The RETURN_AREA must be defined as a 2-byte packed decimal field.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '06'
DATEIN       10   VALUE '150196'
DATEFORM     10   VALUE 'DDMMYY '
DAYLEFT       2   PD               ;Note RETURN_AREA is 2 PD


CALL CONVDATE USING FUNCTION DATEIN DATEFORM DAYLEFT
LIST DATEIN (INPUT DATE) DATEFORM (INPUT DATE'FORMAT)
 DAYLEFT(NUMBER DAYS'LEFT IN MONTH)

T1
T1+50       'COMPUTE DAYS LEFT IN CAL MONTH'

STOP
```

Figure 64 Compute Number of Days Left in a Calendar Month Input

Output

```
                COMPUTE DAYS LEFT IN CAL MONTH
   INPUT DATE      INPUT DATE       NUMBER DAYS
                     FORMAT        LEFT IN MONTH
     150196          DDMMYY             016
```

Figure 65 Compute Number of Days Left in a Calendar Month Output

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION     2      VALUE   '06'
DATEIN       4  BI  VALUE 152367    ; DECEMBER 14, 1999
DATEFORM    10      VALUE 'LLLL '
DAYLEFT      2  PD

CALL CONVDATE USING FUNCTION DATEIN DATEFORM DAYLEFT

LIST DATEIN (INPUT DATE) DATEFORM (INPUT DATE'FORMAT)
     DAYLEFT (NUMBER DAYS'LEFT IN MONTH)
T1
T1+50   'COMPUTE DAYS LEFT IN CAL MONTH'
```

Figure 66 Compute Number of Days Left in a Calendar Month (Lilian Date) Input

Output

```
COMPUTE DAYS LEFT IN CAL MONTH
               INPUT DATE    INPUT DATE       NUMBER DAYS
                               FORMAT        LEFT IN MONTH
                 152367        LLLL             17
```

Figure 67 Compute Number of Days Left in a Calendar Month (Lilian Date) Output

Function 07: Compute Week Ending Date for a Given Date

```
CALL CONVDATE   USING   FUNCTION     DATE_IN   DATE_FORMAT
RETURN_AREA RETURN_FORMAT [WEEKEND_CODE [YEAR_PFX]]
```

The optional parameter WEEKEND_CODE can be used to override the definition in the date table for the day of the week considered to be ending day of the work week. If used, it is a data name (1 character) containing the value 1 though 7 to identify the day (1 = Monday, 7 = Sunday) which will be used as the day ending the work week.

If YEAR_PFX is specified, a value must be specified for WEEKEND_CODE. If no override of the week ending day is wanted, set the WEEKEND_CODE data name to zero.

The RETURN_AREA data name must be a character field of 10 bytes, but the number of bytes returned depends on the RETURN_FORMAT mask.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION     2   VALUE '07'
DATEIN      10   VALUE '15011996'
DAT1FORM    10   VALUE 'DDMMYYYY '
DATEOUT     10
DAT2FORM    10   VALUE 'MMDDYY '

CALL CONVDATE USING FUNCTION DATEIN DAT1FORM DATEOUT DAT2FORM
LIST DATEIN (INPUT DATE) DAT1FORM (INPUT DATE'FORMAT)
 DATEOUT (WEEK ENDING'DATE) DAT2FORM(OUTPUT DATE'FORMAT)

T1
T1+50       'COMPUTE WEEK ENDING DATE'

STOP
```

Figure 68 Compute Week Ending Date for a Given Date Input

Output

```
                    COMPUTE WEEK ENDING DATE
 INPUT DATE      INPUT DATE      WEEK ENDING      OUTPUT DATE
                   FORMAT           DATE             FORMAT
15011996         DDMMYYYY         012196            MMDDYY
```

Figure 69 Compute Week Ending Date for a Given Date Output

Function 08: Adjust a Date Up or Down

```
CALL CONVDATE   USING   FUNCTION  DATE_IN DATE_FORMAT ADJUST
RETURN_AREA RETURN_FORMAT [YEAR_PFX]
```

The ADJUST data name is the number of days to adjust the DATE_IN value. It is a character field 5 bytes long with leading high order zeros for a positive adjustment, or a 6-byte field with a leading negative sign (-) and high order zeros for a negative adjustment.

The RETURN_AREA is a 10-byte character field, although the actual number of characters returned depends on the RETURN_FORMAT mask.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION     2   VALUE '08'
DATEIN      10   VALUE '960115'
DAT1FORM    10   VALUE 'YYMMDD '
ADJUSTMT     6   VALUE '-00014 '
DATEOUT     10
DAT2FORM    10   VALUE 'YYMMDD '


CALL CONVDATE USING FUNCTION DATEIN DAT1FORM ADJUSTMT DATEOUT
```

Figure 70 Adjust a Date Up or Down Input

```
     DAT2FORM
LIST DATEIN (INPUT DATE) DAT1FORM (INPUT DATE'FORMAT)
     ADJUSTMT (ADJUST FACTOR) DATEOUT (OUTPUT DATE)
     DAT2FORM(OUTPUT DATE'FORMAT)

T1
T1+50       'ADJUST DATE UP OR DOWN'

STOP
```

Figure 70 Adjust a Date Up or Down Input

Output

```
                                ADJUST DATE UP OR DOWN
INPUT DATE     INPUT DATE     ADJUST FACTOR     OUTPUT DATE     OUTPUT DATE
                 FORMAT                                           FORMAT
  960115         YYMMDD          -00014            960101         YYMMDD
```

Figure 71 Adjust a Date Up or Down Output

Function 09: Convert a Two-digit Month Code to Alphabetic Month

```
CALL CONVDATE   USING   FUNCTION   DATE_IN  INPUT_FORMAT  RETURN_AREA
```

The RETURN_AREA data name must be a 9-byte character field, to contain the returned alphabetic month.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '09'
DATEIN       10   VALUE '01/15/1996'
DATEFORM     11   VALUE 'MM/DD/YYYY '
DATEOUT       9


CALL CONVDATE USING FUNCTION DATEIN DATEFORM DATEOUT
LIST DATEIN (INPUT DATE) DATEFORM (INPUT DATE'FORMAT)
     DATEOUT (ALPHA MONTH)

T1
T1+50       'CONVERT 2-DIGIT MONTH TO ALPHA'

STOP
```

Figure 72 Convert a Two-digit Month Code to Alphabetic Month Input

Output

```
                CONVERT 2 DIGIT MONTH TO ALPHA
   INPUT DATE     INPUT DATE     ALPHA MONTH
                    FORMAT
   01/15/1996     MM/DD/YYYY        JANUARY
```

Figure 73 Convert a Two-digit Month Code to Alphabetic Month Output

Function 10: Compute Number of Work days Between Two Dates

```
CALL CONVDATE   USING   FUNCTION  DATE1_IN DATE1_FORMAT DATE2_IN
      DATE2_FORMAT RETURN_AREA [YEAR_PFX]
```

The RETURN_AREA data name must be a field defined as 6-byte packed decimal 1.

Input

```
FILE ININ DUMMY

WORKAREA
FUNCTION      2   VALUE '10'
DAT1IN       10   VALUE '00/01/15'
DAT1FORM     10   VALUE 'YY/MM/DD '
DAT2IN       10   VALUE '00031'
DAT2FORM     10   VALUE 'YYDDD '
WORKDAYS      6    PD 1


CALL CONVDATE USING FUNCTION DAT1IN DAT1FORM DAT2IN DAT2FORM WORKDAYS
LIST DAT1IN (FIRST DATE) DAT1FORM (FIRST DATE'FORMAT)
         DAT2IN (SECOND DATE) DAT2FORM (SECOND DATE'FORMAT)
        WORKDAYS (NUMBER OF'WORK DAYS)

T1
T1+50       'COMPUTE #WORK DAYS BET 2 DATES'

STOP
```

Figure 74 Compute Number of Work Days Between Two Dates Input

Output

```
                          COMPUTE #WORK DAYS BET 2 DATES
 FIRST DATE    FIRST DATE    SECOND DATE    SECOND DATE     NUMBER OF
                 FORMAT                        FORMAT       WORK DAYS
   00/01/15      YY/MM/DD        00031          YYDDD           12.5
```

Figure 75 Compute Number of Work Days Between Two Dates Output

Note: If you assign a non-work day as a holiday, the results will be incorrect (such as, if Fridays are normally a half-work day and New Year's Day falls on a Friday, indicate only a half-day off in the holiday and date table).

Function 11: Check for Holiday and Time Off

```
CALL CONVDATE   USING   FUNCTION     DATE_IN  FORMAT_IN  RETURN_AREA [YEAR_PFX]
```

Since all of the holidays in the date table (CONVDATT) must be of the 4-digit year format, the DATE_IN data name should contain a 4-digit year date, and be accordingly identified in the FORMAT_IN data name. If a 2-digit year is used, it will be converted to 4-digit year.

The RETURN_AREA must be defined as a 2-byte packed decimal field, and will have one of the following values:

00 Indicates no time off for the input date.

05 Indicates a half day off.

10 Indicates a full day off or non-work day (weekend).

Note: No checking is performed to determine if the date is a non-work date.

Input

```
FILE ININ DUMMY
WORKAREA
FUNCTION      2   VALUE '11'
DATEIN       10   VALUE '01012001'
DATEFORM     10   VALUE 'MMDDYYYY '
TIMEOFF       2   PD
CALL CONVDATE USING FUNCTION DATEIN DATEFORM TIMEOFF
LIST DATEIN (INPUT DATE) DATEFORM (INPUT DATE'FORMAT)
     TIMEOFF (HOLIDAY AND'TIME OFF)
T1
T1+50       'CHECK FOR HOLIDAY AND TIME OFF'
STOP
```

Figure 76 Check for Holiday and Time Off Input

Output

```
                CHECK FOR HOLIDAY AND TIME OFF
 INPUT DATE      INPUT DATE      HOLIDAY AND
                   FORMAT          TIME OFF
  01012001       MMDDYYYY           010
```

Figure 77 Check for Holiday and Time Off Output

Function 12: Load Another Date Table

```
CALL CONVDATE   USING   FUNCTION     DATETABLE
```

The DATETABLE data name must be an 8-byte character field, and must contain the name, as identified in your phase or load library, of the new date table to be loaded (see CONVDATT on page 184 for creating the standard date table and any additional date tables). All CONVDATE functions from this point on will use the date table loaded by this function 12 call. If this function is not used, the default date table (CONVDATT) is used.

If the newly loaded table is invalid, all subsequent calls to CONVDATE will result in a return code of 11.

```
WORKAREA
FUNCTION      02   VALUE '12'
DATETABLE     08   VALUE 'MYTAB   '

CALL CONVDATE   USING FUNCTION   DATETABLE

STOP
```

Figure 78 Load another Date Table

Function 13: Provide Date of the First Working Date of the Month

```
CALL CONVDATE   USING   FUNCTION DATE_IN FORMAT_IN RETURN_AREA
     FORMAT_RETURN [YEAR_PFX]
```

The RETURN_AREA data name must be defined as character, with a length (maximum of 10) corresponding to the length of the FORMAT_RETURN data name.

If there are no work days in the month, a return code of 13 is set.

Input

```
FILE   ININ   DUMMY

WORKAREA
FUNCTION    2  VALUE '13'
DATEIN     10  VALUE '2001/01/15'
DAT1FORM   11  VALUE 'YYYY/MM/DD '
DATEOUT    10
DAT2FORM   10  VALUE 'MM/DD/YY '


CALL CONVDATE USING FUNCTION DATEIN DAT1FORM DATEOUT DAT2FORM

LIST DATEIN(INPUT DATE) DAT1FORM(INPUT DATE'FORMAT)
         DATEOUT(FIRST WORKING'DAY OF MONTH)
          DAT2FORM(OUTPUT DATE'FORMAT)

T1
T1+50  'FIRST WORKING DAY OF MONTH'
STOP
```

Figure 79 Date of the First Working Date of the Month Input

Output

```
                    FIRST WORKING DAY OF MONTH
INPUT DATE      INPUT DATE      FIRST WORKING      OUTPUT DATE
                  FORMAT        DAY OF MONTH         FORMAT
2001/01/15      YYYY/MM/DD        01/02/01          MM/DD/YY
```

Figure 80 Date of the First Working Date of the Month Output

CONVDATE Error Codes

The CONVDATE routine will provide a return code in Register 15, which can be accessed by your VISION:Results program. You must specify the RETCODE parameter on your DYLINSTL installation macro. (For more information about DYLINSTL, see the *Advantage VISION:Results for z/OS Installation Guide*.) The return codes from CONVDATE are:

0000 Function was performed successfully.

0001 Invalid function code.

0002 Invalid number of parameters for this function.

0003 Date code format invalid for this function.

0004 Address, in parameter list is not valid.

0005 Invalid Julian date.
0006 Invalid calendar date input.
0007 Invalid year input.
0008 Invalid day code.
0009 Invalid date adjustment factor.
0010 Invalid month abbreviation.
0011 Invalid holiday table (user date table).
0012 Invalid optional parameter value.
0013 No work days for this month in the user date table.

DYLFMTJG (Converting Julian or Gregorian Date to a Calendar Date)

DYLFMTJG allows z/OS and VSE users to convert a Julian date (96015) or Gregorian date (011596) to a calendar date (January 15, 1996).

DYLFMTJG is called by using a CALL command. The CALL command requires three parameters that identify the input, output, and error message output areas.

Typical uses of DYLFMTJG include:

- Printing dates on letters
- Printing dates on preprinted forms
- Printing dates in report titles

This subroutine is self-relocating, serially reusable, and less than 2K long.

z/OS: DYLFMTJG must be link edited to a load library. If the module is not linked to an automatically searched library, a JOBLIB or STEPLIB statement must be included in the JCL to identify the library in which DYLFMTJG resides.

VSE: DYLFMTJG must be link edited to a VSE/SP phase library or a VSE core image library.

Note: No checking is performed to determine if the date is a non-work date.

The format of the call is:

```
CALL DYLFMTJG 2K USING [Parm0] Parm1 Parm2 Parm3
```

The following information is passed to subroutine DYLFMTJG:

- Specification that date to be converted contains a four-digit year (optional).
- Format of date to be converted (Julian or Gregorian).
- Date to be converted.

- Area to receive the converted date.
- Area to contain an error message if an error is found.

The CALL command should be coded as follows:

Entry	Description	
DYLFMTJG	Name of called subroutine.	
2K	Memory size for DYLFMTJG (VSE only).	
(parameters)	Designates four areas (parameters) that contain data for the conversion.	
Parm0 (optional)	Identifies a four-byte CH field with a value of FMT4. This indicates that Parm1 contains a date with a four-digit year. If Parm0 is not provided, Parm1 must contain a date in two-digit year format. The century prefix returned in Parm2 depends on the following: ■ If the reserved word DYLCENTRY1 is non-blank, its value is used as the century prefix. ■ If DYLCENTRY1 is blank, the value of CENTNEW determines the century prefix. For a description of CENTNEW and establishing a default value for DYLCENTRY1, see the *Advantage VISION:Results for z/OS Installation Guide*. For a description of DYLCENTRY1, see the *Advantage VISION:Results for z/OS Getting Started*.	
Parm 1 (No Parm0 provided)	Identifies a seven-byte area containing a date with a two-digit year.	
	Byte 1:	Contains the format of the date to be converted: J = Julian (YYDDD), G = Gregorian (MMDDYY).
	Bytes 2-7:	Contain the date to be converted: a five-byte Julian date (YYDDD) or a six-byte Gregorian date (MMDDYY).
Parm1 (Parm0 provided)	Identifies a nine-byte area containing a date with a four-digit year.	
	Byte 1:	Contains the format of the date to be converted: J=Julian (YYYYDDD) or G=Gregorian (MMDDYYY).
	Bytes 2-9:	Contain the date to be converted: a seven-byte Julian date (YYYYDDD) or an eight-byte Gregorian date (MMDDYYYY).
Parm 2	Identifies an 18-byte area.	

Entry	Description	
	Bytes 1-18:	Contain the converted date returned by DYLFMTJG (for example, January 15, 2006).
Parm 3	Identifies an 80-byte area.	
	Byte 1:	Contains E if input is erroneous, blank otherwise.
	Bytes 2-80:	Contain error message if input is erroneous, blank otherwise. (Error message is self-explanatory.)
	Note: A check for E in the first byte of this parameter should be made following return from DYLFMTJG.	

The example shown below converts a Gregorian date to a calendar date.

```
FILE SYSIN CARD
     INDATE         6   1  NU  D  (INPUT'DATE)

WORKAREA
   PARM1            7   1
     INTYPE         1   1         VALUE 'G'
     GREGDATE       6   2

   PARM2           18  11
     CALDATE       18  11         (CALENDAR DATE)

   PARM3           80  21
     ERRMESG       80  21
     ERRFLAG        1  21

MOVE INDATE      TO   GREGDATE
MOVE SPACES      TO   ERRMESG

CALL DYLFMTJG 2K USING PARM1 PARM2 PARM3

IF ERRFLAG  EQ  'E'
   HEXPRINT PARM1
   HEXPRINT PARM2
   PRINT ERRMESG
   REJECT
ENDIF

LIST INDATE  CALDATE
ACCEPT

T1  'CONVERT GREGORIAN OR JULIAN DATE TO CALENDAR DATE'
T2  'DYLFMTJG EXAMPLE'  WITH 2 AFTER

FIN
022898
011596
052591
```

Figure 81 Gregorian Date to a Calendar Date

```
CONVERT GREGORIAN OR JULIAN DATE TO CALENDAR DATE
                DYLFMTJG EXAMPLE

          INPUT         CALENDAR DATE
           DATE

         02/28/98     FEBRUARY 28, 1998
         01/15/96     JANUARY 16, 1996
         05/25/91     MAY 25, 1991
```

Chapter 8: APPC and z/OS

VISION:Results supports cooperative and distributed applications between z/OS systems using APPC and z/OS, which allows a VISION:Results z/OS program to initiate peer-to-peer communications with a remote VISION:Results z/OS program by using LU 6.2 protocols. Also, this allows any program using SAA Common Program Interface Communications (CPI Communications) services on any platform to initiate peer-to-peer communications with a remote z/OS VISION:Results program by using LU 6.2 protocols.

The VISION:Results programs DYLAPPC1 and DYLAPPC2, which are reproduced in this chapter and also supplied on the distribution tape, are examples of how APPC and z/OS might be used with VISION:Results. These programs closely follow examples given in the IBM® manual *MVS and ESA Application Development: Writing Transaction Programs for APPC and MVS*.

These two VISION:Results programs exchange data in a conversation using multi-transaction scheduling. The outbound program, DYLAPPC1, allocates the conversation and sends data. The inbound VISION:Results program, DYLAPPC2, receives this data and sends data back to DYLAPPC1.

DYLAPPC1

Structure

The DYLAPPC1 program includes the following:

- The normal OPTION statement.
- The required FILE SYSIN DUMMY statement.
- The COPY DYLATB statement (DYLATB is included on the distribution tape). DYLATB defines all the necessary declarations for using the APPC and z/OS CALL statements.

Work Areas

- The first work area defined shows all of the values used in the CALL statements to APPC.
- The second work area shows all of the possible return code values from the APPC and z/OS calls.
- The third work area defines areas for the parameters for the APPC and z/OS CALL statements themselves.

Program Execution

Execution code starts after the work areas are defined.

1. The ALLOCATE command is the first command executed. It allocates a session between the outbound program, DYLAPPC1, and the inbound program, DYLAPPC2.
2. Next, the Send Data function is executed. It sends the data from DYLAPPC1 to DYLAPPC2.
3. The third function, Receive and Wait, waits for data to be sent back from DYLAPPC2 and receives that data. When the data is available, the program continues executing.
4. The IF statement checks several “status received” conditions and performs the Confirmed function.
5. The Confirmed function sends a confirmation reply to DYLAPPC2, allowing the two programs to synchronize their processing.
6. The LIST statement displays the data received from DYLAPPC2.
7. The next Send Data function sends another message to DYLAPPC2.
8. The Deallocate function deallocates the conversation from DYLAPPC2 and the program ends.

DYLAPPC2

Structure

The DYLAPPC2 program has a structure similar to DYLAPPC1 which includes the OPTION, FILE, and COPY DYLATB statements.

Program Execution

1. Program execution begins with the Get Transaction function. This function holds consecutive conversations with multiple partner programs without having to be initiated and terminated for each conversation.

2. The Get Conversation function returns the conversation ID which references the conversation to which it is attached. The DOWHILE statements continue to execute as long as there is data to be received from DYLAPPC1. When all the data is received from DYLAPPC1, DYLAPPC2 falls through the ENDDO.
3. Then, if DYLAPPC2 receives the proper status response, it sends a confirmed message to DYLAPPC1 and the Send Data function sends data back to DYLAPPC1.
4. Then DYLAPPC2 performs the Receive and Wait function while waiting for more data from DYLAPPC1.
5. When DYLAPPC2 receives the data from DYLAPPC1, it falls through to DOWHILE and performs another Confirmed function and another Get Transaction.
6. The Get Transaction function waits until it receives more input from another program or until DYLAPPC1 is run again, at which time the DOWHILE loop is executed again. If the default wait time is reached, the program will fall through the DOWHILE loop to perform the Return Transaction function and the program ends.

Other Considerations

In order for DYLAPPC1 and DYLAPPC2 to communicate, VTAM, APPC, and z/OS must be made aware of the existence of DYLAPPC1 and DYLAPPC2. See the appropriate IBM z/OS/ESA manuals for more information.

In addition, IBM supplies two APPC and z/OS service routines, ATBPBI and ATBATP, that must be relink edited to allow VISION:Results to load them from the load library. Relink these routines with an ALIAS statement for each entry point within the routine. For the entry point names in each of these routines, see the IBM manual *MVS/ESA Application Development: Writing Transaction Programs for APPC/MVS*.

Sample VISION:Results Programs

Sample of DYLAPPC1 Program

```
        ****************************************************************    1
        *                                                              *    2
        *                    PGM: DYLAPPC1                             *    3
        *                                                              *    4
        ****************************************************************    5
                                                                             6
        OPTION STRUCTURED2 XREF 60 LONG                                      7
                                                                             8
        FILE SYSIN DUMMY                                                     9
                                                                            10
                                                                            11
        COPY DYLATB                                                         12
COPIED ****************************************************************    13
COPIED *                                                              *    14
COPIED *  DYLATB - INTERFACE DECLARATIONS FOR LU 6.2 PROTOCOL BOUNDARY  *    15
COPIED *           INTERFACES - VISION:RESULTS                        *    16
COPIED *                                                              *    17
COPIED ****************************************************************    18
COPIED                                                                      19
COPIED WORKAREA                                                             20
COPIED                                                                      21
COPIED    ATB_F0        4 BI VALUE 0                                        22
COPIED    ATB_F1        4 BI VALUE 1                                        23
COPIED    ATB_F2        4 BI VALUE 2                                        24
COPIED    ATB_F3        4 BI VALUE 3                                        25
COPIED    ATB_F4        4 BI VALUE 4                                        26
COPIED    ATB_F100      4 BI VALUE 100                                      27
COPIED    ATB_F101      4 BI VALUE 101                                      28
COPIED    ATB_F102      4 BI VALUE 102                                      29
COPIED                                                                      30
COPIED ***   CONVERSION TYPE VALUES                                         31
COPIED                                                                      32
COPIED    REDEF ATB_F0    ATB_BASIC_CONVERSATION         4 BI               33
COPIED    REDEF ATB_F1    ATB_MAPPED_CONVERSATION        4 BI               34
COPIED                                                                      35
COPIED ***   DATA RECEIVED VALUES                                           36
COPIED                                                                      37
COPIED    REDEF ATB_F0    ATB_NO_DATA_RECEIVED           4 BI              3 8
COPIED    REDEF ATB_F1    ATB_DATA_RECEIVED              4 BI               39
COPIED    REDEF ATB_F2    ATB_COMPLETE_RECEIVED          4 BI               40
COPIED    REDEF ATB_F3    ATB_INCOMPLETE_RECEIVED        4 BI               41
COPIED                                                                      42
COPIED ***   DEALLOCATE TYPE VALUES                                         43
COPIED                                                                      44
COPIED    REDEF ATB_F0    ATB_DEALLOCATE_SYNC_LEVEL      4 BI               45
COPIED    REDEF ATB_F1    ATB_DEALLOCATE_FLUSH           4 BI               46
COPIED    REDEF ATB_F2    ATB_DEALLOCATE_CONFIRM         4 BI               47
COPIED    REDEF ATB_F3    ATB_DEALLOCATE_ABEND           4 BI               48
COPIED                                                                      49
COPIED ***   ERROR DIRECTION VALUES                                         50
COPIED                                                                      51
COPIED    REDEF ATB_F0    ATB_RECIEVE_ERROR              4 BI               52
COPIED    REDEF ATB_F1    ATB_SEND_ERROR                 4 BI               53
COPIED                                                                      54
COPIED ***   FILL VALUES                                                    55
COPIED                                                                      56
COPIED    REDEF ATB_F0    ATB_FILL_LL                    4 BI               57
COPIED    REDEF ATB_F1    ATB_FILL_BUFFER                4 BI               58
COPIED                                                                      59
COPIED ***   LOCK VALUES                                                    60
COPIED                                                                      61
COPIED    REDEF ATB_F100  ATB_LOCKS_SHORT                4 BI               62
COPIED    REDEF ATB_F101  ATB_LOCKS_LONG                 4 BI               63
COPIED                                                                      64
```

Figure 82 DYLAPPC1 (Page 1 of 6)

```
COPIED ***   NOTIFY TYPE VALUES                                              65
COPIED                                                                       66
COPIED    REDEF ATB_F0    ATB_NOTIFY_TYPE_NONE           4 BI               67
COPIED    REDEF ATB_F1    ATB_NOTIFY_TYPE_ECB            4 BI               68
COPIED                                                                       69
COPIED ***   PREPARE TO RECEIVE TYPE VALUES                                  70
COPIED                                                                       71
COPIED    REDEF ATB_F0    ATB_PREP_TO_RECEIVE_SYNC_LEVEL 4 BI               72
COPIED    REDEF ATB_F1    ATB_PREP_TO_RECEIVE_FLUSH      4 BI               73
COPIED    REDEF ATB_F2    ATB_PREP_TO_RECEIVE_CONFIRM    4 BI               74
COPIED                                                                       75
COPIED ***   REQUEST TO SEND RECEIVED VALUES                                 76
COPIED                                                                       77
COPIED    REDEF ATB_F0    ATB_REQ_TO_SEND_NOT_RECEIVED   4 BI               78
COPIED    REDEF ATB_F1    ATB_REQ_TO_SEND_RECEIVED       4 BI               79
COPIED                                                                       80
COPIED ***   RETURN CONTROL VALUES                                           81
COPIED                                                                       82
COPIED    REDEF ATB_F0    ATB_WHEN_SESSION_ALLOCATED     4 BI               83
COPIED    REDEF ATB_F1    ATB_IMMEDIATE                  4 BI               84
COPIED    REDEF ATB_F100  ATB_WHEN_CONWINNER_ALLOCATED   4 BI               85
COPIED                                                                       86
COPIED ***   SECURITY TYPE VALUES                                            87
COPIED                                                                       88
COPIED    REDEF ATB_F100  ATB_SECURITY_NONE              4 BI               89
COPIED    REDEF ATB_F101  ATB_SECURITY_SAME              4 BI               90
COPIED    REDEF ATB_F102  ATB_SECURITY_PROGRAM           4 BI               91
COPIED                                                                       92
COPIED ***   SEND TYPE VALUES                                                93
COPIED                                                                       94
COPIED    REDEF ATB_F0    ATB_BUFFER_DATA                4 BI               95
COPIED    REDEF ATB_F1    ATB_SEND_AND_FLUSH             4 BI               96
COPIED    REDEF ATB_F2    ATB_SEND_AND_CONFIRM           4 BI               97
COPIED    REDEF ATB_F3    ATB_SEND_AND_PREP_TO_RECEIVE   4 BI               98
COPIED    REDEF ATB_F4    ATB_SEND_AND_DEALLOCATE        4 BI               99
COPIED                                                                      100
COPIED ***   STATUS RECEIVED VALUES                                         101
COPIED                                                                      102
COPIED    REDEF ATB_F0    ATB_NO_STATUS_RECEIVED         4 BI              103
COPIED    REDEF ATB_F1    ATB_SEND_RECEIVED              4 BI              104
COPIED    REDEF ATB_F2    ATB_CONFIRM_RECEIVED           4 BI              105
COPIED    REDEF ATB_F3    ATB_CONFIRM_SEND_RECEIVED      4 BI              106
COPIED    REDEF ATB_F4    ATB_CONFIRM_DEALLOC_RECEIVED   4 BI              107
COPIED                                                                      108
COPIED ***   SYNC LEVEL VALUES                                              109
COPIED                                                                      110
COPIED    REDEF ATB_F0    ATB_NONE                       4 BI              111
COPIED    REDEF ATB_F1    ATB_CONFIRM                    4 BI              112
COPIED WORKAREA                                                             114
COPIED                                                                      115
COPIED ***   RETURN CODE VALUES                                             116
COPIED                                                                      117
COPIED    ATB_OK                          4 BI VALUE 0                      118
COPIED    ATB_ALLOCATE_FAILURE_NO_RETRY   4 BI VALUE 1                      119
COPIED    ATB_ALLOCATE_FAILURE_RETRY      4 BI VALUE 2                      120
COPIED    ATB_CONVERSATION_TYPE_MISMATCH  4 BI VALUE 3                      121
COPIED    ATB_PIP_NOT_SPECIFIED_CORRECTLY 4 BI VALUE 5                      122
COPIED    ATB_SECURITY_NOT_VALID          4 BI VALUE 6                      123
COPIED    ATB_SYNC_LVL_NOT_SUPPORTED_PGM  4 BI VALUE 8                      124
COPIED    ATB_TPN_NOT_RECOGNIZED          4 BI VALUE 9                      125
COPIED    ATB_TP_NOT_AVAILABLE_NO_ENTRY   4 BI VALUE 10                     126
COPIED    ATB_TP_NOT_AVAILABLE_RETRY      4 BI VALUE 11                     127
COPIED    ATB_DEALLOCATED_ABEND           4 BI VALUE 17                     128
COPIED    ATB_DEALLOCATED_NORMAL          4 BI VALUE 18                     129
COPIED    ATB_PARAMETER_ERROR             4 BI VALUE 19                     130
COPIED    ATB_PRODUCT_SPECIFIC_ERROR      4 BI VALUE 20                     131
COPIED    ATB_PROGRAM_ERROR_NO_TRUNC      4 BI VALUE 21                     132
COPIED    ATB_PROGRAM_ERROR_NO_PURGING    4 BI VALUE 22                     133
COPIED    ATB_PROGRAM_ERROR_TRUNC         4 BI VALUE 23                     134
COPIED    ATB_PROGRAM_PARAMETER_CHECK     4 BI VALUE 24                     135
COPIED    ATB_PROGRAM_STATE_CHECK         4 BI VALUE 25                     136
COPIED    ATB_RESOURCE_FAILURE_NO_RETRY   4 BI VALUE 26                     137
COPIED    ATB_RESOURCE_FAILURE_RETRY      4 BI VALUE 27                     138
COPIED    ATB_UNSUCCESSFUL                4 BI VALUE 28                     139
COPIED    ATB_DEALLOCATED_ABEND_SVC       4 BI VALUE 30                     140
COPIED    ATB_DEALLOCATED_ABEND_TIMER     4 BI VALUE 31                     141
```

Figure 82 DYLAPPC1 (Page 2 of 6)

```
COPIED    ATB_SVC_ERROR_NO_TRUNC          4 BI VALUE 32                    142
COPIED    ATB_SVC_ERROR_PURGING           4 BI VALUE 33                    143
COPIED    ATB_SVC_ERROR_TRUNC             4 BI VALUE 34                    144
COPIED                                                                     145
      ************************************************************************ 146
COPIED *                                                                  * 147
COPIED *                  FIELDS FOR CALL PARAMETERS                      * 148
COPIED *                                                                  * 149
COPIED ************************************************************************ 150
COPIED                                                                     151
COPIED WORKAREA                                                            152
COPIED                                                                     153
COPIED    W_CONVERSATION_TYPE       4 BI                                   154
COPIED    W_SYM_DEST_NAME           8 CH                                   155
COPIED    W_PARTNER_LU_NAME        17 CH                                   156
COPIED    W_MODE_NAME               8 CH                                   157
COPIED    W_TP_NAME_LENGTH          4 BI                                   158
COPIED    W_TP_NAME                64 CH                                   159
COPIED    W_RETURN_CONTROL          4 BI                                   160
COPIED    W_SYNC_LEVEL              4 BI                                   161
COPIED    W_SECURITY_TYPE           4 BI                                   162
COPIED    W_USER_ID                10 CH                                   163
COPIED    W_PASSWORD               10 CH                                   164
COPIED    W_PROFILE                10 CH                                   165
COPIED    W_USER_TOKEN              1 CH                                   166
COPIED    W_CONVERSATION_ID         8 CH                                   167
COPIED    W_CONVERSATION_CORRELATOR 8 CH                                   168
COPIED    W_NOTIFY_TYPE             8 CH                                   169
COPIED    W_TP_ID                   8 CH                                   170
COPIED    W_RETURN_CODE             4 BI                                   171
COPIED    W_RC                      4 BI                                   172
COPIED    W_DEALLOCATE_TYPE         4 BI                                   173
COPIED    W_FILL                    4 BI                                   174
COPIED    W_RECEIVE_LENGTH          4 BI                                   175
COPIED    W_RECEIVE_ACCESS_TOKEN    4 BI                                   176
COPIED    W_RECEIVE_BUFFER         80 CH                                   177
COPIED    W_STATUS_RECEIVED         4 BI                                   178
COPIED    W_DATA_RECEIVED_VALUE     4 BI                                   179
COPIED    W_REQUEST_TO_SEND_VALUE   4 BI                                   180
COPIED    W_SEND_TYPE               4 BI                                   181
COPIED    W_SEND_DATA_LENGTH        4 BI                                   182
COPIED    W_SEND_LENGTH             4 BI                                   183
COPIED    W_SEND_ACCESS_TOKEN       4 BI                                   184
COPIED    W_SEND_BUFFER            80 CH                                   185
COPIED    W_SEND_DATA              78 CH                                   186
                                                                           187
      ************************************************************************ 188
      *                                                                  * 189
      *                           VARIABLES                              * 190
      *                                                                  * 191
      ************************************************************************ 192
                                                                           193
      WORKAREA                                                             194
                                                                           195
         LOW_VALUES            8 CH                                        196
            REDEF LOW_VALUES                                               197
         LOW_VALUES_1          4 BI VALUE 0                                198
         LOW_VALUES_2          4 BI VALUE 0                                199
                                                                           200
                                                                           201
      ************************************************************************ 202
      *                                                                  * 203
      *                    CODE STARTS HERE                              * 204
      *                                                                  * 205
      ************************************************************************ 206
                                                                           207
         PERFORM ALLOCATE  TO ALLOCATE_X                                   208
                                                                           209
         PERFORM SENDDATA1 TO SENDDATA1X                                   210
                                                                           211
         PERFORM REC_WAIT  TO REC_WAIT_X                                   212
                                                                           213
         IF W_STATUS_RECEIVED EQ ATB_CONFIRM_RECEIVED      OR              214
            W_STATUS_RECEIVED EQ ATB_CONFIRM_SEND_RECEIVED OR              215
            W_STATUS_RECEIVED EQ ATB_CONFIRM_DEALLOC_RECEIVED              216
                                                                           217
```

Figure 82 DYLAPPC1 (Page 3 of 6)

```
      PERFORM CONFIRMED TO CONFIRMEDX                                    218
                                                                         219
   ENDIF                                                                 220
                                                                         221
   LIST 'RECEIVE_BUFFER' AT 1 W_RECEIVE_BUFFER AT 30                     222
                                                                         223
   PERFORM SENDDATA2 TO SENDDATA2X                                       224
                                                                         225
   PERFORM DEALLOC   TO DEALLOC_X                                        226
                                                                         227
   STOP                                                                  228
********************************************************************** 230
*                                                                    *  231
*                          ALLOCATE                                  *  232
*                                                                    *  233
********************************************************************** 234
                                                                         235
ALLOCATE:                                                                236
                                                                         237
   W_CONVERSATION_TYPE       = ATB_MAPPED_CONVERSATION                   238
   W_SYM_DEST_NAME           = ' '                                       239
   W_PARTNER_LU_NAME         = 'ACBSV01 '                                240
   W_MODE_NAME               = 'APPCMOD1'                                241
   W_TP_NAME_LENGTH          = 8                                         242
   W_TP_NAME                 = 'DYLAPPC2'                                243
   W_RETURN_CONTROL          = ATB_WHEN_SESSION_ALLOCATED                244
   W_SYNC_LEVEL              = ATB_CONFIRM                               245
   W_SECURITY_TYPE           = ATB_SECURITY_NONE                         246
   W_USER_ID                 = 'IBMUSER'                                 247
   W_PASSWORD                = 'JERRYS'                                  248
   W_PROFILE                 = ' '                                       249
   W_USER_TOKEN              = LOW_VALUES                                250
*  W_CONVERSATION_ID           RETURNED BY ALLOCATE                      251
   W_NOTIFY_TYPE             = LOW_VALUES                                252
   W_TP_ID                   = LOW_VALUES                                253
*  W_RETURN_CODE               RETURNED BY ALLOCATE                      254
                                                                         255
   CALL ATBALLC USING W_CONVERSATION_TYPE                                256
                      W_SYM_DEST_NAME                                    257
                      W_PARTNER_LU_NAME                                  258
                      W_MODE_NAME                                        259
                      W_TP_NAME_LENGTH                                   260
                      W_TP_NAME                                          261
                      W_RETURN_CONTROL                                   262
                      W_SYNC_LEVEL                                       263
                      W_SECURITY_TYPE                                    264
                      W_USER_ID                                          265
                      W_PASSWORD                                         266
                      W_PROFILE                                          267
                      W_USER_TOKEN                                       268
                      W_CONVERSATION_ID                                  269
                      W_NOTIFY_TYPE                                      270
                      W_TP_ID                                            271
                      W_RETURN_CODE                                      272
                                                                         273
   IF W_RETURN_CODE NE ATB_OK STOP ENDIF                                 274
                                                                         275
ALLOCATE_X:                                                              276
********************************************************************** 278
*                                                                    *  279
*                      SEND_DATA_1                                   *  280
*                                                                    *  281
********************************************************************** 282
                                                                         283
SENDDATA1:                                                               284
                                                                         285
*  W_CONVERSATION_ID           SET BY ALLOCATE                           286
   W_SEND_TYPE               = ATB_BUFFER_DATA                           287
   W_SEND_LENGTH             = 80                                        288
   W_SEND_ACCESS_TOKEN       = 0                                         289
   W_SEND_BUFFER             = 'APPC1 - FIRST BUFFER'                    290
*  W_REQUEST_TO_SEND_VALUE     RETURNED BY SEND                          291
   W_NOTIFY_TYPE             = LOW_VALUES                                292
*  W_RETURN_CODE               RETURNED BY SEND                          293
                                                                         294
   CALL ATBSEND USING W_CONVERSATION_ID                                  295
```

Figure 82 DYLAPPC1 (Page 4 of 6)

```
                    W_SEND_TYPE                                     296
                    W_SEND_LENGTH                                   297
                    W_SEND_ACCESS_TOKEN                             298
                    W_SEND_BUFFER                                   299
                    W_REQUEST_TO_SEND_VALUE                         300
                    W_NOTIFY_TYPE                                   301
                    W_RETURN_CODE                                   302
                                                                    303
SENDDATA1X:                                                         304
********************************************************************  306
*                                                                 *  307
*                    RECEIVE_AND_WAIT                             *  308
*                                                                 *  309
********************************************************************  310
                                                                    311
REC_WAIT:                                                           312
                                                                    313
*  W_CONVERSATION_ID           SET BY ALLOCATE                      314
   W_FILL                   = ATB_FILL_LL                           315
   W_RECEIVE_LENGTH         = 80                                    316
   W_RECEIVE_ACCESS_TOKEN   = 0                                     317
*  W_RECEIVE_BUFFER            RETURNED BY RCVW                     318
*  W_STATUS_RECEIVED           RETURNED BY RCVW                     319
*  W_DATA_RECEIVED_VALUE       RETURNED BY RCVW                     320
*  W_REQUEST_TO_SEND_VALUE     RETURNED BY RCVW                     321
   W_NOTIFY_TYPE            = LOW_VALUES                            322
*  W_RETURN_CODE               RETURNED BY RCVW                     323
                                                                    324
   CALL ATBRCVW USING W_CONVERSATION_ID                             325
                    W_FILL                                          326
                    W_RECEIVE_LENGTH                                327
                    W_RECEIVE_ACCESS_TOKEN                          328
                    W_RECEIVE_BUFFER                                329
                    W_STATUS_RECEIVED                               330
                    W_DATA_RECEIVED_VALUE                           331
                    W_REQUEST_TO_SEND_VALUE                         332
                    W_NOTIFY_TYPE                                   333
                    W_RETURN_CODE                                   334
                                                                    335
REC_WAIT_X:                                                         336
                                                                    337
********************************************************************  338
*                                                                 *  339
*                       CONFIRMED                                 *  340
*                                                                 *  341
********************************************************************  342
                                                                    343
CONFIRMED:                                                          344
                                                                    345
*  W_CONVERSATION_ID           SET BY ALLOCATE                      346
   W_NOTIFY_TYPE            = LOW_VALUES                            347
*  W_RETURN_CODE               RETURNED BY CFMD                     348
                                                                    349
   CALL ATBCFMD USING W_CONVERSATION_ID                             350
                    W_NOTIFY_TYPE                                   351
                    W_RETURN_CODE                                   352
                                                                    353
CONFIRMEDX:                                                         354
********************************************************************  356
*                                                                 *  357
*                       SEND_DATA_2                               *  358
*                                                                 *  359
********************************************************************  360
                                                                    361
SENDDATA2:                                                          362
                                                                    363
*  W_CONVERSATION_ID           SET BY ALLOCATE                      364
   W_SEND_TYPE              = ATB_BUFFER_DATA                       365
   W_SEND_LENGTH            = 80                                    366
   W_SEND_ACCESS_TOKEN      = 0                                     367
   W_SEND_BUFFER            = 'APPC1, SECOND BUFFER'                368
*  W_REQUEST_TO_SEND_VALUE     RETURNED BY SEND                     369
   W_NOTIFY_TYPE            = LOW_VALUES                            370
*  W_RETURN_CODE               RETURNED BY SEND                     371
                                                                    372
   CALL ATBSEND USING W_CONVERSATION_ID                             373
```

Figure 82 DYLAPPC1 (Page 5 of 6)

```
                        W_SEND_TYPE                                     374
                        W_SEND_LENGTH                                   375
                        W_SEND_ACCESS_TOKEN                             376
                        W_SEND_BUFFER                                   377
                        W_REQUEST_TO_SEND_VALUE                         378
                        W_NOTIFY_TYPE                                   379
                        W_RETURN_CODE                                   380
                                                                        381
      SENDDATA2X:                                                       382
                                                                        383
      ********************************************************************* 384
      *                                                                 * 385
      *                      DEALLOCATE                                 * 386
      *                                                                 * 387
      ********************************************************************* 388
                                                                        389
      DEALLOC:                                                          390
                                                                        391
      *  W_CONVERSATION_ID           SET BY ALLOCATE                    392
         W_DEALLOCATE_TYPE         = ATB_DEALLOCATE_SYNC_LEVEL          393
         W_NOTIFY_TYPE             = LOW_VALUES                         394
      *  W_RETURN_CODE               RETURNED BY CFMD                   395
                                                                        396
         CALL ATBDEAL USING W_CONVERSATION_ID                           397
                            W_DEALLOCATE_TYPE                           398
                            W_NOTIFY_TYPE                               399
                            W_RETURN_CODE                               400
                                                                        401
      DEALLOC_X:                                                        402
```

Figure 82 DYLAPPC1 (Page 6 of 6)

Sample of DYLAPPC1 – cross references program:

```
C R O S S   R E F E R E N C E
                                 A       T D
                                 R       Y E
            DATANAME             E       P C
             OR TAG         LOCN A  SIZE  E . DEFN            REFERENCES

ALLOCATE_X:                                    276 - 208
ALLOCATE:                                      236 - 208
ATB_BUFFER_DATA               393 R     4 BI 0   95 - 287  365
ATB_CONFIRM                   397 R     4 BI 0  112 - 245
ATB_CONFIRM_DEALLOC_RECEIVED  409 R     4 BI 0  107 - 214
ATB_CONFIRM_RECEIVED          401 R     4 BI 0  105 - 214
ATB_CONFIRM_SEND_RECEIVED     405 R     4 BI 0  106 - 214
ATB_DEALLOCATE_SYNC_LEVEL     393 R     4 BI 0   45 - 393
ATB_FILL_LL                   393 R     4 BI 0   57 - 315
ATB_MAPPED_CONVERSATION       397 R     4 BI 0   34 - 238
ATB_OK                        425 R     4 BI 0  118 - 274
ATB_SECURITY_NONE             413 R     4 BI 0   89 - 246
ATB_WHEN_SESSION_ALLOCATED    393 R     4 BI 0   83 - 244
CONFIRMED:                                     344 - 218
CONFIRMEDX:                                    354 - 218
DEALLOC_X:                                     402 - 226
DEALLOC:                                       390 - 226
LOW_VALUES                   1009 R     8 CH    196 - 250  252  253  292  322  347
                                                    370  394
REC_WAIT_X:                                    336 - 212
REC_WAIT:                                      312 - 212
SENDDATA1:                                     284 - 210
SENDDATA1X:                                    304 - 210
SENDDATA2:                                     362 - 224
SENDDATA2X:                                    382 - 224
W_CONVERSATION_ID             685 R     8 CH    167 - 256  295  325  350  373  397
W_CONVERSATION_TYPE           537 R     4 BI 0  154 - 238  256
W_DATA_RECEIVED_VALUE         825 R     4 BI 0  179 - 325
W_DEALLOCATE_TYPE             725 R     4 BI 0  173 - 393  397
W_FILL                        729 R     4 BI 0  174 - 315  325
W_MODE_NAME                   566 R     8 CH    157 - 241  256
```

Figure 83 DYLAPPC1 – cross references (Page 1 of 2)

```
W_NOTIFY_TYPE                701 R     8 CH    169 - 252  256  292  295  322  325
                                                     347  350  370  373  394  397
W_PARTNER_LU_NAME            549 R    17 CH    156 - 240  256

  C R O S S   R E F E R E N C E
                                 A        T D
                                 R        Y E
            DATANAME             E        P C
             OR TAG         LOCN A  SIZE  E . DEFN            REFERENCES

W_PASSWORD                   664 R    10 CH    164 - 248  256
W_PROFILE                    674 R    10 CH    165 - 249  256
W_RECEIVE_ACCESS_TOKEN       737 R     4 BI 0  176 - 317  325
W_RECEIVE_BUFFER             741 R    80 CH    177 - 222  325
W_RECEIVE_LENGTH             733 R     4 BI 0  175 - 316  325
W_REQUEST_TO_SEND_VALUE      829 R     4 BI 0  180 - 295  325  373
W_RETURN_CODE                717 R     4 BI 0  171 - 256 274  295  325  350  373
                                                     397
W_RETURN_CONTROL             642 R     4 BI 0  160 - 244  256
W_SECURITY_TYPE              650 R     4 BI 0  162 - 246  256
W_SEND_ACCESS_TOKEN          845 R     4 BI 0  184 - 289  295  367  373
W_SEND_BUFFER                849 R    80 CH    185 - 290  295  368  373
W_SEND_LENGTH                841 R     4 BI 0  183 - 288  295  366  373
W_SEND_TYPE                  833 R     4 BI 0  181 - 287  295  365  373
W_STATUS_RECEIVED            821 R     4 BI 0  178 - 214  214  214  325
W_SYM_DEST_NAME              541 R     8 CH    155 - 239  256
W_SYNC_LEVEL                 646 R     4 BI 0  161 - 245  256
W_TP_ID                      709 R     8 CH    170 - 253  256
W_TP_NAME                    578 R    64 CH    159 - 243  256
W_TP_NAME_LENGTH             574 R     4 BI 0  158 - 242  256
W_USER_ID                    654 R    10 CH    163 - 247  256
W_USER_TOKEN                 684 R     1 CH    166 - 250  256
```

Figure 83 DYLAPPC1 – cross references (Page 2 of 2)

Sample of DYLAPPC2 Program

```
      ************************************************************************    1
      *                                                                      *    2
      *                         PGM: DYLAPPC2                                *    3
      *                                                                      *    4
      ************************************************************************    5
                                                                                  6
      OPTION STRUCTURED2 XREF 60 LONG                                             7
                                                                                  8
      FILE SYSIN DUMMY                                                            9
                                                                                 10
                                                                                 11
      COPY DYLATB                                                                12
COPIED ************************************************************************   13
COPIED *                                                                      *   14
COPIED *  DYLATB - INTERFACE DECLARATIONS FOR LU 6.2 PROTOCOL BOUNDARY        *   15
COPIED *           INTERFACES - DYL-280.                                      *   16
COPIED *                                                                      *   17
COPIED ************************************************************************   18
COPIED                                                                            19
COPIED WORKAREA                                                                   20
COPIED                                                                            21
COPIED    ATB_F0        4 BI VALUE 0                                            22
COPIED    ATB_F1        4 BI VALUE 1                                            23
COPIED    ATB_F2        4 BI VALUE 2                                            24
COPIED    ATB_F3        4 BI VALUE 3                                            25
COPIED    ATB_F4        4 BI VALUE 4                                            26
COPIED    ATB_F100      4 BI VALUE 100                                          27
COPIED    ATB_F101      4 BI VALUE 101                                          28
COPIED    ATB_F102      4 BI VALUE 102                                          29
COPIED                                                                            30
COPIED ***   CONVERSION TYPE VALUES                                               31
COPIED                                                                            32
COPIED    REDEF ATB_F0    ATB_BASIC_CONVERSATION         4 BI                     33
COPIED    REDEF ATB_F1    ATB_MAPPED_CONVERSATION        4 BI                     34
COPIED                                                                            35
```

Figure 84 DYLAPPC2 (Page 1 of 7)

```
COPIED ***   DATA RECEIVED VALUES                                          36
COPIED                                                                     37
COPIED    REDEF ATB_F0    ATB_NO_DATA_RECEIVED           4 BI              38
COPIED    REDEF ATB_F1    ATB_DATA_RECEIVED              4 BI              39
COPIED    REDEF ATB_F2    ATB_COMPLETE_RECEIVED          4 BI              40
COPIED    REDEF ATB_F3    ATB_INCOMPLETE_RECEIVED        4 BI              41
COPIED                                                                     42
COPIED ***   DEALLOCATE TYPE VALUES                                        43
COPIED                                                                     44
COPIED    REDEF ATB_F0    ATB_DEALLOCATE_SYNC_LEVEL      4 BI              45
COPIED    REDEF ATB_F1    ATB_DEALLOCATE_FLUSH           4 BI              46
COPIED    REDEF ATB_F2    ATB_DEALLOCATE_CONFIRM         4 BI              47
COPIED    REDEF ATB_F3    ATB_DEALLOCATE_ABEND           4 BI              48
COPIED                                                                     49
COPIED ***   ERROR DIRECTION VALUES                                        50
COPIED                                                                     51
COPIED    REDEF ATB_F0    ATB_RECIEVE_ERROR              4 BI              52
COPIED    REDEF ATB_F1    ATB_SEND_ERROR                 4 BI              53
COPIED                                                                     54
COPIED ***   FILL VALUES                                                   55
COPIED                                                                     56
COPIED    REDEF ATB_F0    ATB_FILL_LL                    4 BI              57
COPIED    REDEF ATB_F1    ATB_FILL_BUFFER                4 BI              58
COPIED                                                                     59
COPIED ***   LOCK VALUES                                                   60
COPIED                                                                     61
COPIED    REDEF ATB_F100  ATB_LOCKS_SHORT                4 BI              62
COPIED    REDEF ATB_F101  ATB_LOCKS_LONG                 4 BI              63
COPIED                                                                     64
COPIED ***   NOTIFY TYPE VALUES                                            65
COPIED                                                                     66
COPIED    REDEF ATB_F0    ATB_NOTIFY_TYPE_NONE           4 BI              67
COPIED    REDEF ATB_F1    ATB_NOTIFY_TYPE_ECB            4 BI              68
COPIED                                                                     69
COPIED ***   PREPARE TO RECEIVE TYPE VALUES                                70
COPIED                                                                     71
COPIED    REDEF ATB_F0    ATB_PREP_TO_RECEIVE_SYNC_LEVEL 4 BI              72
COPIED    REDEF ATB_F1    ATB_PREP_TO_RECEIVE_FLUSH      4 BI              73
COPIED    REDEF ATB_F2    ATB_PREP_TO_RECEIVE_CONFIRM    4 BI              74
COPIED                                                                     75
COPIED ***   REQUEST TO SEND RECEIVED VALUES                               76
COPIED                                                                     77
COPIED    REDEF ATB_F0    ATB_REQ_TO_SEND_NOT_RECEIVED   4 BI              78
COPIED    REDEF ATB_F1    ATB_REQ_TO_SEND_RECEIVED       4 BI              79
COPIED                                                                     80
COPIED ***   RETURN CONTROL VALUES                                         81
COPIED                                                                     82
COPIED    REDEF ATB_F0    ATB_WHEN_SESSION_ALLOCATED     4 BI              83
COPIED    REDEF ATB_F1    ATB_IMMEDIATE                  4 BI              84
COPIED    REDEF ATB_F100  ATB_WHEN_CONWINNER_ALLOCATED   4 BI              85
COPIED                                                                     86
COPIED ***   SECURITY TYPE VALUES                                          87
COPIED                                                                     88
COPIED    REDEF ATB_F100  ATB_SECURITY_NONE              4 BI              89
COPIED    REDEF ATB_F101  ATB_SECURITY_SAME              4 BI              90
COPIED    REDEF ATB_F102  ATB_SECURITY_PROGRAM           4 BI              91
COPIED                                                                     92
COPIED ***   SEND TYPE VALUES                                              93
COPIED                                                                     94
COPIED    REDEF ATB_F0    ATB_BUFFER_DATA                4 BI              95
COPIED    REDEF ATB_F1    ATB_SEND_AND_FLUSH             4 BI              96
COPIED    REDEF ATB_F2    ATB_SEND_AND_CONFIRM           4 BI              97
COPIED    REDEF ATB_F3    ATB_SEND_AND_PREP_TO_RECEIVE   4 BI              98
COPIED    REDEF ATB_F4    ATB_SEND_AND_DEALLOCATE        4 BI              99
COPIED                                                                    100
COPIED ***   STATUS RECEIVED VALUES                                       101
COPIED                                                                    102
COPIED    REDEF ATB_F0    ATB_NO_STATUS_RECEIVED         4 BI             103
COPIED    REDEF ATB_F1    ATB_SEND_RECEIVED              4 BI             104
COPIED    REDEF ATB_F2    ATB_CONFIRM_RECEIVED           4 BI             105
COPIED    REDEF ATB_F3    ATB_CONFIRM_SEND_RECEIVED      4 BI             106
COPIED    REDEF ATB_F4    ATB_CONFIRM_DEALLOC_RECEIVED   4 BI             107
COPIED                                                                    108
```

Figure 84 DYLAPPC2 (Page 2 of 7)

```
COPIED ***   SYNC LEVEL VALUES                                                  109
COPIED                                                                          110
COPIED    REDEF ATB_F0    ATB_NONE                      4 BI                  111
COPIED    REDEF ATB_F1    ATB_CONFIRM                   4 BI                  112
COPIED WORKAREA                                                                 114
COPIED                                                                          115
COPIED ***   RETURN CODE VALUES                                                 116
COPIED                                                                          117
COPIED    ATB_OK                          4 BI VALUE 0                          118
COPIED    ATB_ALLOCATE_FAILURE_NO_RETRY   4 BI VALUE 1                          119
COPIED    ATB_ALLOCATE_FAILURE_RETRY      4 BI VALUE 2                          120
COPIED    ATB_CONVERSATION_TYPE_MISMATCH  4 BI VALUE 3                          121
COPIED    ATB_PIP_NOT_SPECIFIED_CORRECTLY 4 BI VALUE 5                          122
COPIED    ATB_SECURITY_NOT_VALID          4 BI VALUE 6                          123
COPIED    ATB_SYNC_LVL_NOT_SUPPORTED_PGM  4 BI VALUE 8                          124
COPIED    ATB_TPN_NOT_RECOGNIZED          4 BI VALUE 9                          125
COPIED    ATB_TP_NOT_AVAILABLE_NO_ENTRY   4 BI VALUE 10                         126
COPIED    ATB_TP_NOT_AVAILABLE_RETRY      4 BI VALUE 11                         127
COPIED    ATB_DEALLOCATED_ABEND           4 BI VALUE 17                         128
COPIED    ATB_DEALLOCATED_NORMAL          4 BI VALUE 18                         129
COPIED    ATB_PARAMETER_ERROR             4 BI VALUE 19                         130
COPIED    ATB_PRODUCT_SPECIFIC_ERROR      4 BI VALUE 20                         131
COPIED    ATB_PROGRAM_ERROR_NO_TRUNC      4 BI VALUE 21                         132
COPIED    ATB_PROGRAM_ERROR_NO_PURGING    4 BI VALUE 22                         133
COPIED    ATB_PROGRAM_ERROR_TRUNC         4 BI VALUE 23                         134
COPIED    ATB_PROGRAM_PARAMETER_CHECK     4 BI VALUE 24                         135
COPIED    ATB_PROGRAM_STATE_CHECK         4 BI VALUE 25                         136
COPIED    ATB_RESOURCE_FAILURE_NO_RETRY   4 BI VALUE 26                         137
COPIED    ATB_RESOURCE_FAILURE_RETRY      4 BI VALUE 27                         138
COPIED    ATB_UNSUCCESSFUL                4 BI VALUE 28                         139
COPIED    ATB_DEALLOCATED_ABEND_SVC       4 BI VALUE 30                         140
COPIED    ATB_DEALLOCATED_ABEND_TIMER     4 BI VALUE 31                         141
COPIED    ATB_SVC_ERROR_NO_TRUNC          4 BI VALUE 32                         142
COPIED    ATB_SVC_ERROR_PURGING           4 BI VALUE 33                         143
COPIED    ATB_SVC_ERROR_TRUNC             4 BI VALUE 34                         144
COPIED                                                                          145
      ************************************************************************  146
COPIED *                                                                      *  147
COPIED *                  FIELDS FOR CALL PARAMETERS                          *  148
COPIED *                                                                      *  149
COPIED ************************************************************************  150
COPIED                                                                          151
```

Figure 84 DYLAPPC2 (Page 3 of 7)

```
COPIED WORKAREA                                                          152
COPIED                                                                   153
COPIED    W_CONVERSATION_TYPE       4 BI                                 154
COPIED    W_SYM_DEST_NAME           8 CH                                 155
COPIED    W_PARTNER_LU_NAME        17 CH                                 156
COPIED    W_MODE_NAME               8 CH                                 157
COPIED    W_TP_NAME_LENGTH          4 BI                                 158
COPIED    W_TP_NAME                64 CH                                 159
COPIED    W_RETURN_CONTROL          4 BI                                 160
COPIED    W_SYNC_LEVEL              4 BI                                 161
COPIED    W_SECURITY_TYPE           4 BI                                 162
COPIED    W_USER_ID                10 CH                                 163
COPIED    W_PASSWORD               10 CH                                 164
COPIED    W_PROFILE                10 CH                                 165
COPIED    W_USER_TOKEN              1 CH                                 166
COPIED    W_CONVERSATION_ID         8 CH                                 167
COPIED    W_CONVERSATION_CORRELATOR 8 CH                                 168
COPIED    W_NOTIFY_TYPE             8 CH                                 169
COPIED    W_TP_ID                   8 CH                                 170
COPIED    W_RETURN_CODE             4 BI                                 171
COPIED    W_RC                      4 BI                                 172
COPIED    W_DEALLOCATE_TYPE         4 BI                                 173
COPIED    W_FILL                    4 BI                                 174
COPIED    W_RECEIVE_LENGTH          4 BI                                 175
COPIED    W_RECEIVE_ACCESS_TOKEN    4 BI                                 176
COPIED    W_RECEIVE_BUFFER         80 CH                                 177
COPIED    W_STATUS_RECEIVED         4 BI                                 178
COPIED    W_DATA_RECEIVED_VALUE     4 BI                                 179
COPIED    W_REQUEST_TO_SEND_VALUE   4 BI                                 180
COPIED    W_SEND_TYPE               4 BI                                 181
COPIED    W_SEND_DATA_LENGTH        4 BI                                 182
COPIED    W_SEND_LENGTH             4 BI                                 183
COPIED    W_SEND_ACCESS_TOKEN       4 BI                                 184
COPIED    W_SEND_BUFFER            80 CH                                 185
COPIED    W_SEND_DATA              78 CH                                 186
                                                                         187
      ************************************************************       188
      *                                                          *       189
      *                        VARIABLES                         *       190
      *                                                          *       191
      ************************************************************       192
                                                                         193
      WORKAREA                                                           194
                                                                         195
         LOW_VALUES            8 CH                                      196
            REDEF LOW_VALUES                                             197
         LOW_VALUES_1          4 BI VALUE LOWVALUES                      198
         LOW_VALUES_2          4 BI VALUE LOWVALUES                      199
                                                                         200
                                                                         201
      ************************************************************       202
      *                                                          *       203
      *                  CODE STARTS HERE                        *       204
      *                                                          *       205
      ************************************************************       206
                                                                         207
         PERFORM GETTRANS             TO GETTRANS_X                    208
                                                                         209
```

Figure 84 DYLAPPC2 (Page 4 of 7)

```
   DOWHILE W_RETURN_CODE EQ ATB_OK                                          210
                                                                            211
      PERFORM GETCONV           TO GETCONVX                                 212
                                                                            213
      DOWHILE W_STATUS_RECEIVED NE ATB_SEND_RECEIVED AND                    214
              W_RETURN_CODE     EQ ATB_OK                                   215
                                                                            216
         PERFORM REC_WAIT       TO REC_WAIT_X                               217
                                                                            218
      ENDDO                                                                 219
                                                                            220
      IF W_STATUS_RECEIVED EQ ATB_CONFIRM_RECEIVED      OR                  221
         W_STATUS_RECEIVED EQ ATB_CONFIRM_SEND_RECEIVED OR                  222
         W_STATUS_RECEIVED EQ ATB_CONFIRM_DEALLOC_RECEIVED                  223
                                                                            224
         PERFORM CONFIRMED TO CONFIRMEDX                                    225
                                                                            226
      ENDIF                                                                 227
                                                                            228
      LIST 'RECEIVE_BUFFER' AT 1 W_RECEIVE_BUFFER AT 30                     229
                                                                            230
      PERFORM SENDDATA  TO SENDDATAX                                        231
                                                                            232
      DOWHILE W_STATUS_RECEIVED NE ATB_CONFIRM_DEALLOC_RECEIVED AND         233
              W_RETURN_CODE     EQ ATB_OK                                   234
                                                                            235
         PERFORM REC_WAIT       TO REC_WAIT_X                               236
                                                                            237
      ENDDO                                                                 238
                                                                            239
      LIST 'RECEIVE_BUFFER' AT 1 W_RECEIVE_BUFFER AT 30                     240
                                                                            241
      PERFORM CONFIRMED         TO CONFIRMEDX                               242
                                                                            243
      PERFORM GETTRANS          TO GETTRANS_X                               244
                                                                            245
   ENDDO                                                                    246
                                                                            247
   PERFORM RETTRANS             TO RETTRANSX                                248
                                                                            249
   STOP                                                                     250
                                                                            251
*****************************************************************         252
*                                                               *         253
*                       GET_TRANS                               *         254
*                                                               *         255
*****************************************************************         256
                                                                            257
GETTRANS:                                                                   258
                                                                            259
*  W_RETURN_CODE               RETURNED BY GET_TRANS                        260
                                                                            261
   CALL ATBGTRN USING W_RETURN_CODE                                         262
                                                                            263
GETTRANS_X:                                                                 264
*****************************************************************         266
*                                                               *         267
*                     GET_CONVERSATION                          *         268
*                                                               *         269
*****************************************************************         270
                                                                            271
GETCONV:                                                                    272
                                                                            273
*  W_CONVERSATION_ID           RETURNED BY GET                              274
*  W_CONVERSATION_TYPE         RETURNED BY GET                              275
*  W_PARTNER_LU_NAME           RETURNED BY GET                              276
*  W_MODE_NAME                 RETURNED BY GET                              277
*  W_SYNC_LEVEL                RETURNED BY GET                              278
*  W_CONVERSATION_CORRELATOR   RETURNED BY GET                              279
*  W_RETURN_CODE               RETURNED BY GET                              280
                                                                            281
```

Figure 84 DYLAPPC2 (Page 5 of 7)

```
   CALL ATBGETC USING W_CONVERSATION_ID                              282
                      W_CONVERSATION_TYPE                            283
                      W_PARTNER_LU_NAME                              284
                      W_MODE_NAME                                    285
                      W_SYNC_LEVEL                                   286
                      W_CONVERSATION_CORRELATOR                      287
                      W_RETURN_CODE                                  288
                                                                     289
GETCONVX:                                                            290
                                                                     291
*********************************************************************  292
*                                                                   *  293
*                     RECEIVE_AND_WAIT                              *  294
*                                                                   *  295
*********************************************************************  296
                                                                     297
REC_WAIT:                                                            298
                                                                     299
*  W_CONVERSATION_ID           SET BY GET                            300
   W_FILL                   = ATB_FILL_LL                            301
   W_RECEIVE_LENGTH         = 80                                     302
   W_RECEIVE_ACCESS_TOKEN   = 0                                      303
*  W_RECEIVE_BUFFER            RETURNED BY RCVW                      304
*  W_STATUS_RECEIVED           RETURNED BY RCVW                      305
*  W_DATA_RECEIVED_VALUE       RETURNED BY RCVW                      306
*  W_REQUEST_TO_SEND_VALUE     RETURNED BY RCVW                      307
   W_NOTIFY_TYPE            = LOW_VALUES                             308
*  W_RETURN_CODE               RETURNED BY RCVW                      309
                                                                     310
   CALL ATBRCVW USING W_CONVERSATION_ID                              311
                      W_FILL                                         312
                      W_RECEIVE_LENGTH                               313
                      W_RECEIVE_ACCESS_TOKEN                         314
                      W_RECEIVE_BUFFER                               315
                      W_STATUS_RECEIVED                              316
                      W_DATA_RECEIVED_VALUE                          317
                      W_REQUEST_TO_SEND_VALUE                        318
                      W_NOTIFY_TYPE                                  319
                      W_RETURN_CODE                                  320
                                                                     321
REC_WAIT_X:                                                          322
*********************************************************************  324
*                                                                   *  325
*                       CONFIRMED                                   *  326
*                                                                   *  327
*********************************************************************  328
                                                                     329
CONFIRMED:                                                           330
                                                                     331
*  W_CONVERSATION_ID           SET BY GET                            332
   W_NOTIFY_TYPE            = LOW_VALUES                             333
*  W_RETURN_CODE               RETURNED BY CFMD                      334
                                                                     335
   CALL ATBCFMD USING W_CONVERSATION_ID                              336
                      W_NOTIFY_TYPE                                  337
                      W_RETURN_CODE                                  338
                                                                     339
CONFIRMEDX:                                                          340
                                                                     341
*********************************************************************  342
*                                                                   *  343
*                       SEND_DATA                                   *  344
*                                                                   *  345
*********************************************************************  346
                                                                     347
```

Figure 84 DYLAPPC2 (Page 6 of 7)

```
SENDDATA:                                                               348
                                                                        349
*  W_CONVERSATION_ID           SET BY ALLOCATE                          350
   W_SEND_TYPE               = ATB_BUFFER_DATA                          351
   W_SEND_LENGTH             = 80                                       352
   W_SEND_ACCESS_TOKEN       = 0                                        353
   W_SEND_BUFFER             = 'APPC2 - ONLY BUFFER'                    354
*  W_REQUEST_TO_SEND_VALUE     RETURNED BY SEND                         355
   W_NOTIFY_TYPE             = LOW_VALUES                               356
*  W_RETURN_CODE               RETURNED BY SEND                         357
                                                                        358
   CALL ATBSEND USING W_CONVERSATION_ID                                 359
                      W_SEND_TYPE                                       360
                      W_SEND_LENGTH                                     361
                      W_SEND_ACCESS_TOKEN                               362
                      W_SEND_BUFFER                                     363
                      W_REQUEST_TO_SEND_VALUE                           364
                      W_NOTIFY_TYPE                                     365
                      W_RETURN_CODE                                     366
                                                                        367
SENDDATAX:                                                              368
************************************************************************ 370
*                                                                      * 371
*                   RETURN_TRANSACTION                                 * 372
*                                                                      * 373
************************************************************************ 374
                                                                        375
RETTRANS:                                                               376
                                                                        377
*  W_RETURN_CODE               RETURNED BY SEND                         378
                                                                        379
   CALL ATBRTRN USING W_RETURN_CODE                                     380
                                                                        381
RETTRANSX:                                                              382
```

Figure 84 DYLAPPC2 (Page 7 of 7)

Sample of DYLAPPC2 – Cross References Program

```
 C R O S S   R E F E R E N C E
                                A        T D
                                R        Y E
            DATANAME            E        P C
             OR TAG         LOCN A  SIZE  E . DEFN            REFERENCES

ATB_BUFFER_DATA              393 R     4 BI 0   95 - 351
ATB_CONFIRM_DEALLOC_RECEIVED 409 R     4 BI 0  107 - 221  233
ATB_CONFIRM_RECEIVED         401 R     4 BI 0  105 - 221
ATB_CONFIRM_SEND_RECEIVED    405 R     4 BI 0  106 - 221
ATB_FILL_LL                  393 R     4 BI 0   57 - 301
ATB_OK                       425 R     4 BI 0  118 - 210  214  233
ATB_SEND_RECEIVED            397 R     4 BI 0  104 - 214
CONFIRMED:                                     330 - 225  242
CONFIRMEDX:                                    340 - 225  242
GETCONV:                                       272 - 212
GETCONVX:                                      290 - 212
GETTRANS_X:                                    264 - 208  244
GETTRANS:                                      258 - 208  244
LOW_VALUES                  1009 R     8 CH    196 - 308  333  356
REC_WAIT_X:                                    322 - 217  236
REC_WAIT:                                      298 - 217  236
RETTRANS:                                      376 - 248
RETTRANSX:                                     382 - 248
SENDDATA:                                      348 - 231
SENDDATAX:                                     368 - 231
W_CONVERSATION_CORRELATOR    693 R     8 CH    168 - 282
W_CONVERSATION_ID            685 R     8 CH    167 - 282  311  336  359
W_CONVERSATION_TYPE          537 R     4 BI 0  154 - 282
W_DATA_RECEIVED_VALUE        825 R     4 BI 0  179 - 311
W_FILL                       729 R     4 BI 0  174 - 301  311
W_MODE_NAME                  566 R     8 CH    157 - 282
W_NOTIFY_TYPE                701 R     8 CH    169 - 308  311  333  336  356  359
```

Figure 85 DYLAPPC2 – cross references (Page 1 of 2)

```
W_PARTNER_LU_NAME            549 R    17 CH    156 - 282
W_RECEIVE_ACCESS_TOKEN       737 R     4 BI 0  176 - 303  311
W_RECEIVE_BUFFER             741 R    80 CH    177 - 229  240  311
W_RECEIVE_LENGTH             733 R     4 BI 0  175 - 302  311
W_REQUEST_TO_SEND_VALUE      829 R     4 BI 0  180 - 311  359
W_RETURN_CODE                717 R     4 BI 0  171 - 210  214  233  262  282  311
                                                      336  359  380
W_SEND_ACCESS_TOKEN          845 R     4 BI 0  184 - 353  359
W_SEND_BUFFER                849 R    80 CH    185 - 354  359
W_SEND_LENGTH                841 R     4 BI 0  183 - 352  359
W_SEND_TYPE                  833 R     4 BI 0  181 - 351  359
W_STATUS_RECEIVED            821 R     4 BI 0  178 - 214  221  221  221  233  311
W_SYNC_LEVEL                 646 R     4 BI 0  161 - 282
```

Figure 85 DYLAPPC2 – cross references (Page 2 of 2)

Chapter 9: Label Generation

The DYLABEL COPY function generates labels. The number of labels up (to be printed across the page) depends on the label size and the width of the printer. It is your responsibility to determine how many labels are to be produced across a page. The following figure shows one possible label format.

```
TORRES,ERNESTO          CHO PYUNG,SUH           S.F.MEM.HOSP.
23444 PARK LANE         33333 PALL MALL         6789 OLD MAN RD
LOS ANGELES CA          GLENDALE CA 91206       L.A. CA 90023


SANTA FE HOSP ASSN      GENVARDI,G J            TODIPE,MICHAEL
1212 WISCONSIN DRIVE    9785 FLEET STR          5678 REGENT SQR
LOS ANGELES CA 90023    LOS ANGELES CA 90017    INGLEWOOD CA
```

Figure 86 Example of a Label Format

Syntax

DYLABEL is started by using the COPY command.

```
COPY DYLABEL   #1=aaaaaaaaaa  #2=bbbbbbbbbb  #3=cc  #4=ddd
               #5=eeeee       #6=fffffffff   #7=ggggggggg
               #8=hhhhhhhhh   #9=iiiiiiiii   #A=jjjjjjjjj
               #B=kkkkkkkkk   #C=lllllllll   #D=mmmmmmmmm
               #E=nnnnnnnnn   #F=ooooooooo   #G=pppppppppp
```

Figure 87 DYLABEL

where:

#1 Specifies the number of labels to be produced across the page. It can be a numeric value or the data name of a field containing the number of labels to be printed across the page. If a data name is specified, you must define it in your program as a non-character field (NU, PD, or BI) and it must contain the number of labels to be produced across the page. It is your responsibility to make sure that the number of labels requested across the page is valid (for example, it is not more than can fit across the page). If #1 is not specified on the COPY statement, the default is 4-up labels.

#2 Specifies the number of print lines on each label. It can be a numeric value or the data name of a field containing the number of lines to be generated for each label. If a data name is specified, you must define it in your program as a non-character field (NU, PD, or BI) and it must contain the number of lines to be printed on each label. (The number of print lines must be between 1 and 9.) If you do not specify a valid number of lines, an error message is printed at execution time. If #2 is not specified on the COPY command, a default of 6 is assumed.

#3 Specifies the width of each label. This must be a 1- to 3-digit number. The maximum value for #3 is the width of the report (such as, #4). If the value specified for #3 is greater than the width of the report, an error message is printed at execution time. If #3 is not specified on the COPY command, a default of 33 is assumed. #1 times #3 should equal #4.

#4 Specifies the width of the form for the labels. This must be a 1- to 3-digit number. The maximum allowable value is 204 for z/OS or 132 for VSE. If #4 is not specified on the COPY command, a default of 132 is assumed. Specifying an invalid value for #4 results in a validation error message on the VISION:Results REPORT statement.

#5 Specifies the spacing after the last line of the label is printed. Allowable values are 0 to 9 or EJECT. If #5 is not specified, the default is 1, or no blank line after the last line of the label is printed before printing the first line of the next set of labels. Enter one more than the number of blank lines needed to get to the first line of the next label.

#6 Specifies the selection interval for selecting data to produce labels. It can be a numeric value or the data name of a field containing a value n. Every nth record is selected for label printing. If a data name is specified, it must be defined in the program as a non-character field (NU, PD, BI) and must contain a valid number. If #6 is not specified on the COPY command, the assumed interval is 1 (select every record for label printing).

#7 Specifies the maximum number of labels to be produced. It can be a numeric value or the data name of a field containing the number of labels to be produced. If a data name is specified, it must be defined in the program as a non-character field (NU, PD, BI) and must contain a valid number. If #7 is not specified on the COPY command, the assumed default is no limit on the number of labels to be produced.

#8 Specifies what is to be printed as line 1 of the label. It can be a literal or a data name.

#9 Specifies what is to be printed as line 2 of the label. It can be a literal or a data name.

#A Specifies what is to be printed as line 3 of the label. It can be a literal or a data name.

#B Specifies what is to be printed as line 4 of the label. It can be a literal or a data name.

#C Specifies what is to be printed as line 5 of the label. It can be a literal or a data name.

#D Specifies what is to be printed as line 6 of the label. It can be a literal or a data name.

#E Specifies what is to be printed as line 7 of the label. It can be a literal or a data name.

#F Specifies what is to be printed as line 8 of the label. It can be a literal or a data name.

#G Specifies what is to be printed as line 9 of the label. It can be a literal or a data name.

#8 through #G have no defaults. If data names are specified for #8 through #G, you must define them in the program.

Coding Requirements

You must provide the following to use DYLABEL:

- A FILE definition:

  For z/OS
  ```
  FILE filename
  ```

  For VSE
  ```
  FILE filename SYSnnn device recfm recsiz
  e blksize
  ```

  The SYSnnn, device type, record format, record size, and block size must be coded for VSE.

- Field definitions.

  Field definitions must be specified for all data names specified in the COPY command.

- The COPY command.

  You must start DYLABEL using the VISION:Results COPY command and specify the options required.

- User Logic.

  Optionally, you can place your own logic ahead of the COPY command that invokes DYLABEL. This logic can include a SORT, or your own selection logic, or logic to manipulate the incoming data before it is used in the labels. You should not code logic following the COPY command.

The following VISION:Results statements generate the label format illustrated at the beginning of this chapter.

VISION:Results z/OS Example

```
COPY ARDEF
COPY DYLABEL #1=3    #2=3      #3=26     #4=78     #5=3
                               #8=NAME   #9=ADD1   #A=ADD2
FIN
```

Figure 88 Example of VISION:Results z/OS

VISION:Results VSE Example

```
COPY D.ARDEF
COPY D.DYLABEL #1=3     #2=3     #3=26     #4=78     #5=3
                                 #8=NAME   #9=ADD1   #A=ADD2
FIN
```

Figure 89 Example of VISION:Results VSE

DYLABEL COPY Function Example

```
$IF FREE
$DEFAULT       #1=4     #2=6     #3=33     #4=132
               #5=NULL            #6=1      #7=NULL
$DEND
*#1 - NUMBER OF LABELS ACROSS THE PAGE - A NUMERIC VALUE OR
*     DATA NAME, LIMIT IS BASED ON THE LABEL SIZE & REPORT WIDTH
*#2 - NUMBER OF PRINT LINES PER LABEL - A NUMERIC VALUE
*     OR DATA NAME, LIMIT IS 9
*#3 - SIZE OF LABEL - A NUMERIC VALUE, LIMIT IS THE REPORT WIDTH
*#4 - WIDTH OF PRINTER - A NUMERIC VALUE, LIMIT IS 204
*#5 - PAGE EJECT OR SPACE 0-9 LINES AFTER PRINTING LAST LINE
*    - 'EJECT' OR NUMERIC VALUES OF 0 THROUGH 9
*#6 - EVERY NTH RECORD IS SELECTED - A NUMERIC VALUE OR DATA NAME
*#7 - LIMIT THE LABELS - A NUMERIC VALUE OR DATA NAME
*    - MAXIMUM IS 999,999,999
*#8 - LINE 1 FIELD - A DATA NAME OR A LITERAL
*#9 - LINE 2 FIELD - A DATA NAME OR A LITERAL
*#A - LINE 3 FIELD - A DATA NAME OR A LITERAL
*#B - LINE 4 FIELD - A DATA NAME OR A LITERAL
*#C - LINE 5 FIELD - A DATA NAME OR A LITERAL
*#D - LINE 6 FIELD - A DATA NAME OR A LITERAL
*#E - LINE 7 FIELD - A DATA NAME OR A LITERAL
*#F - LINE 8 FIELD - A DATA NAME OR A LITERAL
*#G - LINE 9 FIELD - A DATA NAME OR A LITERAL
WORKAREA
  DYL#ZSELNO  2  PD   VALUE 0
  DYL#ZSZLBL  2  PD   VALUE 0
  DYL#ZLIMIT  5  PD   VALUE 0
  DYL#ZLBLCT  2  PD   VALUE 0
  DYL#ZLINES  2  PD   VALUE 9
  DYL#ZERR   30       VALUE 'CHECK YOUR VARIABLE INPUT(S) !'
  WORKAREA
$IF  #8
   DYL#ZLINE1  #4
$IFE
$IF  #9
  DYL#ZLINE2  #4
$IFE
$IF  #A
  DYL#ZLINE3  #4
$IFE
$IF  #B
  DYL#ZLINE4  #4
$IFE
$IF  #C


DYL#ZLINE5  #4
$IFE
$IF  #D
  DYL#ZLINE6  #4
$IFE
$IF  #E
  DYL#ZLINE7  #4
$IFE
$IF  #F
  DYL#ZLINE8  #4
$IFE
$IF  #G
```

Figure 90 Example of DYLABEL COPY Function (Page 1 of 4)

```
  DYL#ZLINE9  #4
$IFE
$IF  #8
  REDEFINE DYL#ZLINE1 DYL#ZL1LBL #3
$IFE
$IF  #9
  REDEFINE DYL#ZLINE2 DYL#ZL2LBL #3
$IFE
$IF  #A
  REDEFINE DYL#ZLINE3 DYL#ZL3LBL #3
$IFE
$IF  #B
  REDEFINE DYL#ZLINE4 DYL#ZL4LBL #3
$IFE
$IF  #C
  REDEFINE DYL#ZLINE5 DYL#ZL5LBL #3
$IFE
$IF  #D
  REDEFINE DYL#ZLINE6 DYL#ZL6LBL #3
$IFE
$IF  #E
  REDEFINE DYL#ZLINE7 DYL#ZL7LBL #3
$IFE
$IF  #F
  REDEFINE DYL#ZLINE8 DYL#ZL8LBL #3
$IFE
$IF  #G
  REDEFINE DYL#ZLINE9 DYL#ZL9LBL #3
$IFE
*
ON ONE   MOVE #3  TO  DYL#ZSZLBL
         IF DYL#ZLINES LT #2
$IF  #6
         OR DYL#ZSELNO EQ #6
$IFE
$IF  #7
         OR DYL#ZLIMIT EQ #7

$IFE
         OR DYL#ZLIMIT EQ #1
         OR DYL#ZSZLBL GT #4
         OR DYL#ZSZLBL EQ 0
            PRINT DYL#ZERR STOP ENDIF
ENDONE
*
$IF  #6
  DYL#ZSELNO = DYL#ZSELNO + 1
  IF DYL#ZSELNO LT #6 REJECT ENDIF
  DYL#ZSELNO = 0
$IFE
*****************************************************************
*   SAVE DATA OF RECORD(S) IN WORKAREA                          *
*****************************************************************
$IF  #8
  MOVE #8 TO DYL#ZL1LBL (INW)
$IFE
$IF  #9
  MOVE #9 TO DYL#ZL2LBL (INW)
$IFE
$IF  #A
  MOVE #A TO DYL#ZL3LBL (INW)
$IFE
$IF  #B
  MOVE #B TO DYL#ZL4LBL (INW)
$IFE
$IF  #C
  MOVE #C TO DYL#ZL5LBL (INW)
$IFE
$IF  #D
  MOVE #D TO DYL#ZL6LBL (INW)
$IFE
$IF  #E
  MOVE #E TO DYL#ZL7LBL (INW)
$IFE
$IF  #F
  MOVE #F TO DYL#ZL8LBL (INW)
```

Figure 90 Example of DYLABEL COPY Function (Page 2 of 4)

```
$IFE
$IF  #G
  MOVE #G TO DYL#ZL9LBL (INW)
$IFE
*
$IF  #7
  DYL#ZLIMIT = DYL#ZLIMIT + 1
  IF DYL#ZLIMIT LT #7 GOTO DYL#ZCONT ENDIF
  PERFORM DYL#ZPROD TO DYL#ZPRODX STOP
$IFE

*****************************************************************
*   PRINT THE LABELS WHEN THE RIGHT AMOUNT IS ACCUMULATED       *
*****************************************************************
DYL#ZCONT:
  INW = INW + #3
  DYL#ZLBLCT = DYL#ZLBLCT + 1
  IF DYL#ZLBLCT LT #1 ACCEPT ENDIF
  PERFORM DYL#ZPROD TO DYL#ZPRODX
  DYL#ZLBLCT = 0
  INW = 0
  INX = #2 * #4
  MOVE SPACES TO DYL#ZLINE1 LENGTH INX ACCEPT
*
*REPORT #4 WIDE
DYL#ZPROD:
$IF  #8
  LIST DYL#ZLINE1 AT 1
$IFE
$IF  #9
  LIST DYL#ZLINE2 AT 1
$IFE
$IF  #A
  LIST DYL#ZLINE3 AT 1
$IFE
$IF  #B
  LIST DYL#ZLINE4 AT 1
$IFE
$IF  #C
  LIST DYL#ZLINE5 AT 1
$IFE
$IF  #D
  LIST DYL#ZLINE6 AT 1
$IFE
$IF  #E
  LIST DYL#ZLINE7 AT 1
$IFE
$IF  #F
  LIST DYL#ZLINE8 AT 1
$IFE
$IF  #G
  LIST DYL#ZLINE9 AT 1
$IFE
$IF  #5
  WITH #5 AFTER
$IFE
```

Figure 90 Example of DYLABEL COPY Function (Page 3 of 4)

```
DYL#ZPRODX:
  ON FINAL
  $IF  #8
    LIST DYL#ZLINE1 AT 1
  $IFE
  $IF  #9
    LIST DYL#ZLINE2 AT 1
  $IFE
  $IF  #A
    LIST DYL#ZLINE3 AT 1
  $IFE
  $IF  #B
    LIST DYL#ZLINE4 AT 1
  $IFE
  $IF  #C
    LIST DYL#ZLINE5 AT 1
  $IFE
  $IF  #D
    LIST DYL#ZLINE6 AT 1
  $IFE
  $IF  #E
    LIST DYL#ZLINE7 AT 1
  $IFE
  $IF  #F
    LIST DYL#ZLINE8 AT 1
  $IFE
  $IF  #G
    LIST DYL#ZLINE9 AT 1
  $IFE
  $IF  #5
    WITH #5 AFTER
  $IFE
  $IF FIXED
```

Figure 90 Example of DYLABEL COPY Function (Page 4 of 4)

Chapter 10: VISION:Pds

The VISION:Pds facility has been replaced with the native PDS support on a FILE statement and the READDIR and WRITEDIR and READMEM and WRITEMEM commands. The discussion in this chapter is included for compatibility with programs still using the VISION:Pds product.

VISION:Pds is an optional product that allows you to access a z/OS partitioned data set (PDS). You can selectively extract members and directory entries and return them to VISION:Results for interrogation and printing.

A significant advantage of VISION:Pds is the flexibility to format the resulting output in a manner convenient to you. The examples at the end of this chapter demonstrate several techniques of formatting the report.

Some useful applications of VISION:Pds include:

- Updating procedure libraries.
- Listing load module attributes in a specified format.
- Converting libraries.
- Listing members of PROCLIB, MACLIB, and so forth, in member name sequence.
- Backing up PROCLIB and MACLIB to tape.

When VISION:Pds is started by VISION:Results, a parameter list is passed to DYLPDS that describes the functions to be performed and identifies the VISION:Results areas to which the member records and directory entries are to be returned. Member or directory records are returned to the specified VISION:Results area one record at a time. The member and directory records can be examined at this time for possible further processing by VISION:Results.

DYLPDS is called through a CALL command from a VISION:Results procedure logic statement. The parameter list, consisting of three parameters, is provided at this time.

```
CALL DYLPDS USING OPTCODE DIR MEM
```

Parameters

Parm 1 In the above example the first parameter, OPTCODE, identifies the VISION:Results work area, from which the description of the type of retrieval is specified. DYLPDS returns the data set name, volume serial numbers, and logical record length of the accessed PDS to this area.

Parm 2 The second parameter, DIR, provides an area to hold directory records as they are returned to VISION:Results. If the DALL function is used this parameter can also be employed to pass additional control information to DYLPDS (see DALL on page 237).

The user must interrogate the directory entries returned to VISION:Results, parameter 2, for an end of file. When an end of file is reached, DYLPDS places a value of X`FF' in the first byte of the area described by parameter 2.

Parm 3 The third parameter, MEM, describes a VISION:Results data area to which member records from either source or load libraries are returned.

If an error is detected while attempting to execute DYLPDS, an error message is placed in the area specified by the third parameter.

PARAMETER	MINIMUM SIZE	FUNCTION
1	52 bytes	retrieval codes
2	75 bytes	directory records (member name and other data) status indicator
3	80 bytes	member records/error messages

Figure 91 DYLPDS Results

JCL Requirements

In addition to the regular JCL statements required to process a VISION:Results program, an additional DD statement is needed to describe the partitioned data set to be accessed. This statement is //SYSPDS.

```
//SYSPDS DD DSN=pds.dsname,VOL=SER=xxxxxx,DISP=SHR,UNIT=uuu
```

Example:

```
//CA JOB
//JOBLIB   DD DSN=DYLLI,DISP=SHR
//STEP1 EXEC PGM=DYL280
//SYSPRINT DD SYSOUT=A
//SYS004   DD  UNIT=SYSDA,SPACE=(TRK,5)
//SYS280R  DD SYSOUT=A
//SYSPDS DD DSN=dsname,DISP=SHR,UNIT=uuu,VOL=SER=xxxxxx
//SYSIN DD *
  VISION:Results statements
END
/*
```

Figure 92 SYSPDS Example

FILE Statement Requirements

There is no VISION:Results FILE statement that corresponds to the partitioned data set, because DYLPDS is handling the retrieval of the PDS. However, to fulfill the VISION:Results requirement that there be at least one file defined in a run, provide the following:

```
FILE SYSIN CARDS DUMMY
```

MSEL Function

The MSEL (Member Select) option retrieves selected members of a z/OS PDS and returns them, along with corresponding directory entries, to VISION:Results. The function, MSEL, and member names to be selected, along with other control information, is provided to DYLPDS in parameter 1. The parameter list also identifies the VISION:Results areas to which data is returned.

Parameter Requirements

```
PARAMETER 1 MINIMUM SIZE = 52 BYTES
PARAMETER 2 MINIMUM SIZE = 75 BYTES
PARAMETER 3 MINIMUM SIZE = 80 BYTES
```

Use caution to ensure an area of sufficient size is allocated when accessing load libraries. You should specify the largest possible block anticipated. For example, if the PDS is on a 3330, the load library is probably blocked 13030; your third parameter should define an area at least 13030 bytes long.

Parameter Passed to DYLPDS

Parameter 1

```
WORKAREA
  OPTCODE   52
  FUNCTION   4  1  VALUE 'MSEL'
  LETOPT     1  5  VALUE 'L'
  ESTNO      1  6  VALUE 1
  FILLER     2
  MEM       27  9  VALUE 'ACBGEN,ASMFC,ASMFCL,ASMFCLG'
```

Figure 93 Parameter 1 Passed to DYLPDS

Location	Length	Description and Use
1	4	MSEL — Identifies the operation.
5	1	LET Option — Determines if processing is to be halted if a specified member name is not located. ø = (Default) — Place an error message in parameter 3 and terminate processing. L = Continue execution until all members are processed.
6	1	Member Estimate — 1-9 estimate of number of members to be selected. This field establishes the amount of memory to be allocated to DYLPDS to hold directory entries. All values are multiples of 10 (10x74=740 bytes). Default is 2; enough memory for 20 members.
7-8	2	Reserved.
9-nn	Variable	Member Names — A list of member names separated by commas and delimited by a space following the last member. Members are processed in sequence by name; not necessarily in the order given. A maximum of 90 members can be processed.

```
OPTCODE....MSELL1øøACBGEN,ASMFC,ASMFCL,ASMFCLG
```

For this example, parameter 1 is supplied from a work area called OPTCODE. This directs DYLPDS to search a partitioned data set, identified by a SYSPDS DD statement, for members named ACBGEN, ASMFC, ASMFCL, and ASMFCLG. If ACBGEN is not found, processing continues (L in position 5); if position 5 was left blank, processing would terminate if ACBGEN was not found. The 1 in position 6 indicates that there will be ten or fewer members with the listed names.

Parameters Returned from DYLPDS

Parameter 1

```
DSNAME    44   1       ;DATA SET NAME
VOLID      6  45       ;VOLUME SERIAL NUMBER
RECLEN     2  51  BI   ;RECORD LENGTH
```

Figure 94 Parameter 1 From DYLPDS

The field definitions above are redefines of the work area OPTCODE.

Location	Length	Description
1	44	Fully qualified data set name.
45	6	Volume serial number of the volume holding the PDS.
51	2 (Binary)	Logical record length of the member records.

Parameter 2

```
WORKAREA
  DIRECTORY   75
  STATUSCODE   1   1
  DIRENTRY    74   2
```

Figure 95 Parameter 2 From DYLPDS

Location	Length	Description
1	1	Status Indicator — Description of the current processing condition: N = New member to be processed. ø = Processing the current member. E = Error detected. X`FF = End of file. '
2	74	Directory Entry Data — See PDS Directory Format on page 255 for a detailed description.

Parameter 3

```
WORKAREA
  MEMRECORD   80
  ERRORMSG     1  1
```

Figure 96 Parameter 3 From DYLPDS

Location	Length	Description
1	80-nn	Member Record.
1	80	Error Message — If byte 1 of parameter 2 is an E.

MALL

The MALL (ALL MEMBERS) option causes all member records and directory information from a PDS to be retrieved and returned to VISION:Results. Directory records are returned to parameter 2 and member records are returned to parameter 3.

Parameter Requirements

```
Parameter 1 MINIMUM SIZE = 52 BYTES
Parameter 2 MINIMUM SIZE = 75 BYTES
Parameter 3 MINIMUM SIZE = 80 BYTES FOR SOURCE AND
        PROCEDURE LIBRARIES
```

Parameter Passed to DYLPDS

Parameter 1

```
WORKAREA
  OPTCODE    52
  FUNCTION    4  1  VALUE 'MALL'
```

Figure 97 Parameter 1 Passed to DYLPDS

Location	Length	Description
1	4	MALL — Informs DYLPDS of the operation to perform.
5	47	Reserved.

Parameters Returned from DYLPDS

Parameter 1

```
DSNAME   44   1
VOLID     6  45
RECLEN    2  51  BI
```

Figure 98 Parameter 1 From DYLPDS

Location	Length	Description
1	44	Fully qualified data set name.
45	6	Volume serial number of the volume holding the PDS.
51	2 (Binary)	Logical record length of the member records.

Parameter 2

```
WORKAREA
  DIR          75
  STATUSCODE    1  1
  DIRENTRY     74  2
```

Figure 99 Parameter 2 From DYLPDS

Location	Length	Description
1	01	Status Indicator — Description of the current processing condition: N = New member to be processed. ø = Processing the current member. E = Error detected. X`FF' = End of file.
2	74	Directory Entry Data — See PDS Directory Format on page 255 for a detailed description.

Parameter 3

```
WORKAREA
  MEMRECORD   80
  ERRORMSG     1   1
```

Figure 100 Parameter 3 From DYLPDS

Location	Length	Description
1	80-nn	Member Record.
1	80	Error Message — If byte 1 of parameter 2 is an E.

DALL

The DALL (Directory All) option retrieves all directory entries from a z/OS PDS and returns them to VISION:Results for further processing. The user can also have member records passed to the calling VISION:Results program. This option is particularly useful when you want to select and print all members whose name begins with a certain letter combination.

Parameter Requirements

```
PARAMETER 1 MINIMUM SIZE = 52 BYTES
PARAMETER 2 MINIMUM SIZE = 75 BYTES
PARAMETER 3 MINIMUM SIZE = 80 BYTES
```

Use caution to ensure an area of sufficient size is allocated when accessing load libraries. You should specify the largest possible block anticipated. For example, if the PDS is on a 3330, the load library is probably blocked 13030; your third parameter should define an area at least 13030 bytes long.

Parameters Passed to DYLPDS

Parameter 1

```
WORKAREA
  OPTCODE    52
  FUNCTION    4  1  VALUE 'DALL'
```

Figure 101 Parameter 1 Passed to DYLPDS

Location	Length	Description
1	4	DALL — Indicates the function to be performed.
5	47	Reserved.

Parameter 2

```
WORKAREA
  DIRECTORY    75
  PASSOPT       1  1  VALUE 'P'
```

Figure 102 Parameter 2 Passed to DYLPDS

Location	Length	Description
1	1	P = Pass the member records; if coded, the directory can be scanned and selected member records returned.

Parameters Returned from DYLPDS

Parameter 1

```
DSNAME    44   1
  VOLID    6  45
  RECLEN   2  51  BI
```

Figure 103 Parameter 1 From DYLPDS

The above is a redefine of the work area OPTCODE.

Location	Length	Description
1	44	Fully qualified data set name.
45	6	Volume serial number of the volume holding the PDS.
51	2 (Binary)	Logical record length of the member records.

Parameter 2

```
WORKAREA
  DIR           75
  STATUSCODE     1  1
  DIRENTRY      74  2
```

Figure 104 Parameter 2 From DYLPDS

Location	Length	Description
1	1	Status Indicator — Description of the current processing condition: N = New member to be processed. ø = Processing the current member. E = Error detected. X`F' = End of file.
2	74	Directory Entry Data — See PDS Directory Format on page 255 for a detailed description.

Parameter 3

```
WORKAREA
  MEMRECORD     80
  ERRORMSG       1  1
```

Figure 105 Parameter 3 From DYLPDS

Location	Length	Description
1	80-nn	Member Record — If a P was passed to DYLPDS from parameter 2.
1	80	Error Message — If byte 1 of parameter 2 is an E.

The P option applies only to the current member being processed. If an end of file is detected, the pass option is turned off. You must reset the P option for any future members.

The first byte of parameter 2 always contains an N while DYLPDS is performing the DALL function.

Considerations for Use

- Up to 16 input PDSs can be concatenated.
- You must terminate the run by issuing the STOP command. This is because VISION:Results is not reading the PDS and cannot terminate by the regular method (when the input file has reached end of file).
- DYLPDS can issue only one access command (MALL, MSEL, DALL) per job step.
- DYLPDS cannot access variable length records.

Error Messages and Abends

If an error is detected by DYLPDS, an E is placed in byte 1 of parameter 2 and the actual error message is in parameter 3. Position 1 of parameter 2 should be examined after each record for an E (see Print Three Members — MSEL With LET Option on page 242 for the technique).

Error Messages

Error	Cause
ER00 INVALID FUNCTION. MUST BE MALL, DALL, OR MSEL.	A value other than MALL, DALL, or MSEL was passed to DYLPDS from positions 1-4 of parameter 1.
ER01 MEMBER NAME REQUESTED IS GREATER THAN 8 .	A member name being passed to DYLPDS from position 9-nn of parameter 1 of the MSEL option exceeds 8 characters.
ER02 NO MEMBER NAME FOUND ON MSEL: CHECK POSITION 9.	The MSEL option requires a member name in position 9 of parameter 1; it did not find one.
ER03 MEMBER NOT FOUND: LET OPTION NOT SPECIFIED.	The member name specified in parameter 1 was not found in the indicated PDS. If multiple member names are being selected the LET option should be used; if not, DYLPDS terminates after the first unsuccessful directory search.

Error	Cause
ER04 SYSPDS DD STATEMENT MISSING.	The JCL for this job step did not include a //SYSPDS DD statement describing the PDS to be accessed. This is in lieu of an S0C1.
ER06 CHECK PARM 1 POS. 6; # OF ENTRIES TOO S MALL.	Position 6 of parameter 1, for the MSEL option, did not establish enough memory to hold all the directory entries retrieved; increase value and rerun.
ER07 PERM. I/O ERR DETECTED IN ATTEMPT TO SEARCH DIRECTORY.	A permanent I/O error occurred while attempting to read the directory of the PDS.

Abend Codes

Abend	Cause
U3997	A parameter list passed to DYLPDS. You must supply three parameters, separated by at least one space, to DYLPDS on the calling statement.
U3998	DYLPDS was called after an end of file was detected. DYLPDS detected an end of file and returned an end-of-file indicator, X`FF,' to byte 1 of parameter 2. An appropriate exit was not taken at this time.
U3999	DYLPDS was called after an error had been detected. When DYLPDS encounters an error, it places an E in byte 1 of parameter 2 and sets a deactivation switch making DYLPDS ineligible to be called again within this job step.
S0C4 (or S0C5)	The member record returned in parameter 3 exceeded the number of bytes allocated. The area allocated must be at least equal to the largest anticipated block size.

Examples

The following examples illustrate some of the potential uses of DYLPDS. A description and flowchart of procedure logic is included with Examples 1 through 3. The VISION:Results statements are not necessarily described in the order they are encountered in the input stream.

After the narrative portion for each example the SYSOUT of the described example is provided.

Example 1 Print Three Members — MSEL With LET Option

In this example we print members OPEN, CLOSE, and GETMAIN in their entirety from SYS1.MACLIB. No directory information is included in the report.

Step by Step

1. Use a SYSPDS DD statement to specify which partitioned data set to access. The FILE SYSIN CARDS DUMMY does not define the PDS. It satisfies the VISION:Results requirement for the definition of at least one file in the program.

```
//SYSPDS DD DSN=SYS1.MACLIB,DISP=SHR
```

2. DYLPDS is called through a procedure statement CALL command. The parameter list defines the VISION:Results data areas that are used to supply input to and receive output from DYLPDS.

```
DYL: CALL DYLPDS USING OPTCODE DIR MEM
```

3. Parameter 1 defines the area that supplies the DYLPDS function code and options, and the VISION:Results work area OPTCODE in this example.

```
WORKAREA
  OPTCODE 52
  FUNCTION 4 1 VALUE 'MSEL'
  LETOPT 1 5 VALUE 'L'
  MEM 18 9 VALUE 'OPEN,CLOSE,GETMAIN'
```

   The DYLPDS function MSEL is supplied in the work area FUNCTION. The L in LETOPT indicates the LET option. If the member OPEN is not found, the job will not terminate, but instead search for the member CLOSE.

   To print, the members are listed starting in the VALUE clause for MEMBER (MEM) separated with commas.

4. Parameter 2 defines the area to which DYLPDS returns a 1-byte status indicator (STATUSCODE) plus the member name and directory information. This area, DIRECTORY, in this example, was established as a 75-byte work area.

```
WORKAREA
  DIRECTORY 75
  STATUSCODE 1 1
```

5. The first byte of the DIRECTORY area (STATUSCODE) should be examined for an error or end of file indicator.

```
IF STATUSCODE EQ 'E' GOTO ERR ENDIF
IF STATUSCODE EQ X'FF' GOTO EOJ ENDIF
```

   If an error is detected in byte 1 of DIRECTORY (STATUSCODE), control is passed to the statement ERR. The second procedure statement tests for a hexadecimal FF, indicating an end of file has been reached. If an end of file is reached, control is passed to the statement with the tag name EOJ. (The ERR and EOJ statements are discussed later.)

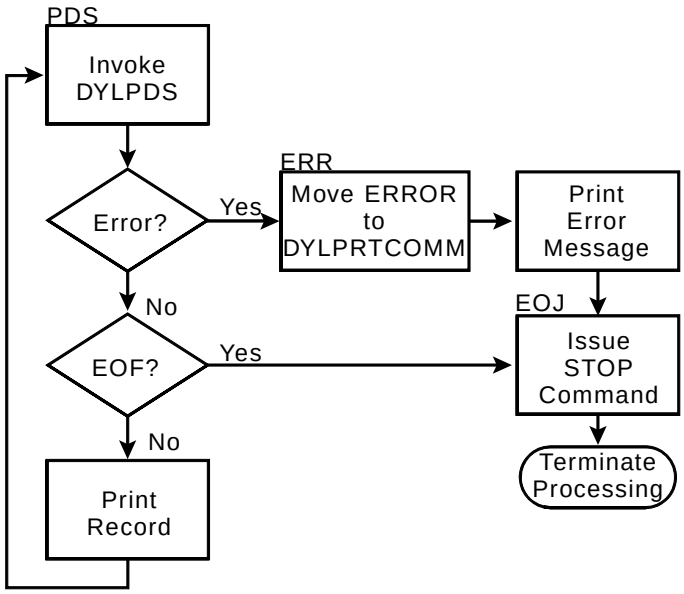

Figure 106 Example 1 Procedure Logic Flowchart

6. If neither of the tests is true, a report line prints and the program branches unconditionally to PDS, retrieving the next member record.

```
PERFORM LISTDET TO LISTEND
  GOTO PDS
```

7. The record to be printed is located in the area specified by parameter 3, MEMRECORD in this example; this area also holds error messages, if any.

```
ERR: MOVE 'ERROR?' TO DYLPRTCOMM
  PRINT MEMRECORD
EOJ: STOP
```

Note: If all members of SYS1.MACLIB are needed, use the MALL option.

If an error is detected, print the comment ERROR? to the left of the record to be printed. This is done by moving the literal ERROR? to the print comments reserved word DYLPRTCOMM and executing a print immediate command (PRINT). The error message is printed from MEMRECORD.

If an end of file is detected, issue a STOP command to terminate the job.

```
FILE INFILE DUMMY

WORKAREA
  OPTCODE     52   1    FUNCTION     4   1      VALUE 'MSEL'
  LETOPT       1   5      VALUE 'L'    MEM         18   9      VALUE 'GETMAIN,CLOSE,OPE
N'
  DSNAME      44   1    VOLID        6  45    RECLEN       2  51  BI

WORKAREA
  DIR         75   1    STATUSCODE   1   1
  DIRENTRY    74   2    MEMNAME      8   2

WORKAREA
```

Figure 107 Example 1 VISION:Results Code (Page 1 of 2)

```
  MEMRECORD   80   1    ERRORMSG     1   1

CONTROL MEMNAME
 DYL:
  CALL DYLPDS USING OPTCODE DIR MEMRECORD
  IF STATUSCODE EQ 'E'
    GOTO ERR
  ENDIF
  IF STATUSCODE EQ X'FF'
    GOTO EOJ
  ENDIF

  PERFORM LISTDET TO LISTEND
  GOTO PDS

ERR:
  MOVE 'ERROR?' TO DYLPRTCOMM
  PRINT MEMRECORD

EOJ:
  STOP

LISTDET:
  LIST SUPPRESS MEMNAME (MEMBER NAME)
               MEMRECORD (CONTENTS OF MEMBER)
LISTEND:

ON CHANGE IN MEMNAME
  MOVE 99 TO DYLLINE

T1 'COMPUTER ASSOCIATES, TEST DYLPDS—EXAMPLE 1' WITH 2 AFTER
```

Figure 107 Example 1 VISION:Results Code (Page 2 of 2)

```
COMPUTER ASSOCIATES, TEST DYLPDS—EXAMPLE 1
 MEMBER NAME                                        CONTENTS OF MEMBER

  CLOSE                    MACRO
                &CNAME     CLOSE &PAR1,&MF=I,&TYPE=,&MODE=
                .*
                .*CLOSE MACRO
                .*
                .*$MAC(CLOSE):
                .*
                .*  COPYRGHT   TYPE=ASMMAC,
                .*             CLASS=RST,
                .*             LEVEL=(HDP2230),
                .*             SECSTMT='IBM INTERNAL USE ONLY'
                .*
                .* STATUS = MVS/XA DFP RELEASE 2.3
                .*
                .*CHANGE ACTIVITY - AS FOLLOWS:
                .*
                .*$L1=ACB31BIT,HDP2230,,STLSS: O/C/E SUPPORT OF 31-BIT VSAM
                .*$01=OY05946,HDP2230,,NSDRJV: EXTRA INSTRUCTION WITH MODE=31
                .*$02=OY07793,HDP2230,870720,STLSS: REMOVE SPLEVEL DEPENDENCE
                .*
                .* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                .*
                         LCLA  &ACTR,&ASUM,&CTR,&CTR1,&LGTH,&NUMBER
                         LCLB  &MFESW,&MFISW,&MFI2SW,&NULLSW,&B(5),&MSW
                         LCLC  &PARA
                .*
                &NUMBER  SETA  N'&PAR1
                .*
                &CTR     SETA  1
                &CTR1    SETA  2
                         AIF   (N'&SYSLIST LE 1).TYPE   IF ONLY 1 (PARMLIST)
```

Figure 108 Example 1 Output (Page 1 of 2)

```
          IHBERMAC 238                    EXCESSIVE POS. PARMS
.TYPE     AIF   (T'&TYPE NE 'O' AND '&TYPE' NE 'T').ERROR8
          AIF   (T'&MODE EQ 'O').MFTEST
          AIF   ('&MODE' NE '31' AND '&MODE' NE '24').ERROR9
          AIF   ('&MODE' EQ '24').MFTEST
&MSW      SETB  1                          SET MODE=31 SWITCH
.MFTEST   AIF   ('&MF' EQ 'L').RTEL
          AIF   ('&MF' EQ 'I').TESTI
          AIF   (N'&MF LE 1).ERROR3
          AIF   ('&MF(1)' NE 'E').ERROR3
&CNAME    IHBINNRA &MF(2)
.*--------------------------------
          AIF   ('&PAR1' EQ '').RGFLIP
.LOOPA    AIF   ('&PAR1(&NUMBER)' NE '').LOOPB
&NUMBER   SETA  &NUMBER-1
          AIF   (&NUMBER EQ 0).RGFLIP
          AGO   .LOOPA
.LOOPB    AIF   ('&PAR1(&CTR)' NE '').TESTCTR
&NULLSW   SETB  1
```

Figure 108 Example 1 Output (Page 2 of 2)

Example 2 List All Member Names of a PDS — DALL Function

This example lists only the member names of partitioned data set Z01.DY.DYLMAC. The names are printed ten across on the report.

The DALL function of DYLPDS retrieves the members directory entries.

Step by Step

In this example, the following VISION:Results data areas pass parameters to and receive directory records from DYLPDS.

Parameter	Data Area		Use
1	OPTCODE	To DYLPDS: From DYLPDS:	Supply function code. Data set name volume serial logical record length.
2	DIRECTORY	To DYLPDS: From DYLPDS:	N/A. Status indicator and directory record.
3	MEMRECORD	To DYLPDS: From DYLPDS:	N/A. Error messages.

1. DYLPDS is called through the CALL command; this statement also provides a parameter list indicating which data areas are utilized.

   ```
   DYL: CALL DYLPDS USING OPTCODE DIR MEMRECORD
   ```

2. DYLPDS returns a status indicator to the first byte of parameter 2 (STATUSCODE). This byte determines the next operation.

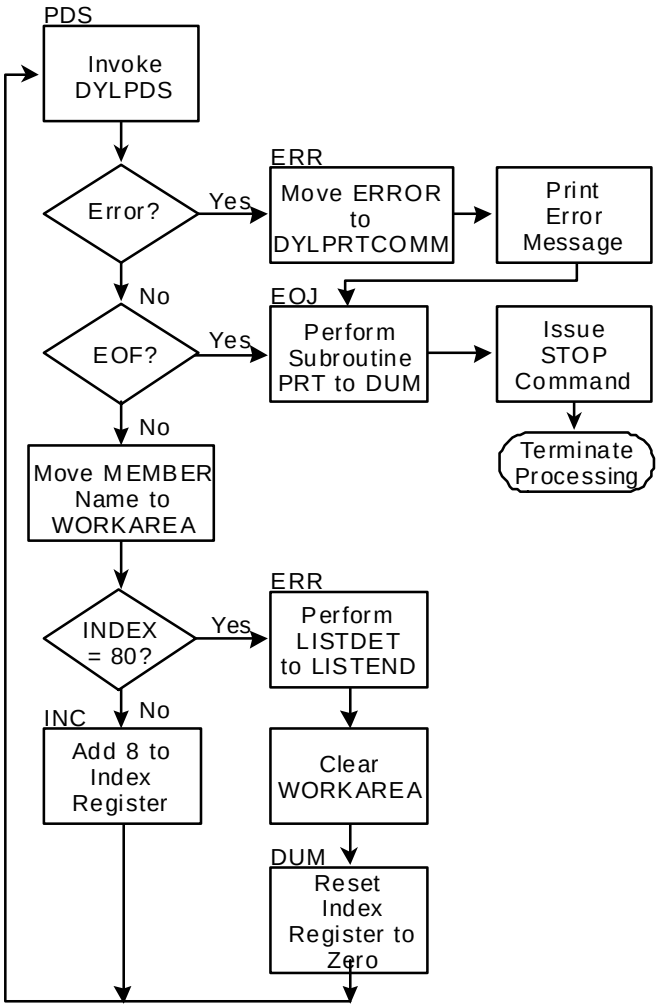

Figure 109 Example 2 Procedure Logic Flowchart

3. The following two statements interrogate STATUSCODE for an error or an end of file, respectively.

```
IF STATUSCODE EQ 'E' GOTO ERROR ENDIF
IF STATUSCODE EQ X'FF' GOTO EOF ENDIF
```

4. This example uses the VISION:Results indexing feature to place the member names in a work area. This is done so that you can print the member names ten across on the report.

5. The next two statements move the member name from MEMNAME to the work area WMEMNAME and check the index counter (INW) to determine if a print line has been filled. If 10 member names are in the work area WMEMNAME, a branch to PRT is taken, and the line is printed.

```
MOVE MEMNAME TO WMEMNAME (INW)
IF INW EQ 80 GOTO PRT ENDIF
```

6. If INW is less than 80, 8 is added to the index counter INW and a branch to DYL is taken to start DYLPDS and retrieve the next directory record.

```
INC: INW=INW+8 GOTO DYL
```

7. The next four statements comprise the report print phase of the program.

```
PRT:
  PERFORM LISTDET TO LISTEND
  MOVE SPACES TO WMEMNAME
DUM:
```

8. After the report line is printed, the index counter is reset. The following statement clears the INW counter to a binary 00 and performs an unconditional branch to PDS to retrieve the next directory record.

```
CLR: MOVE 00 TO INW GOTO PDS
```

9. If an end-of-file condition is detected, a branch to statement EOF takes place. This statement issues a PERFORM command to print any remaining numbers being held in the print line.

```
EOF: PERFORM PRT TO DUM
```

10. The logic executes statements PRT through DUM and returns to the next statement in sequence (EOJ). The tag DUM is used to identify an exit point from the subroutine.

11. The remaining statements cause the comment ERROR? to print to the left of the error message. The PRINT MEMRECORD causes the error message in MEMRECORD and the comment ERROR? to be printed. The last statement issues the STOP command that causes termination at the conclusion of this cycle.

```
ERR:  MOVE 'ERROR?' TO DYLPRTCOMM
      PRINT MEMRECORD
EOF:  PERFORM PRT TO DUM
EOJ:  STOP
```

```
FILE SYSIN CARDS DUMMY
WORKAREA
  OPTCODE 52  FUNCTION 4 1 VALUE 'DALL'  DSNAME 44 1
  VOLID 6 45  RECLEN 2 51 BI
WORKAREA
  DIR 75  STATUSCODE 1 1  DIRENTRY 74 2  MEMNAME 8  2
WORKAREA
  MEMRECORD 80  ERRORMSG 1 1
WORKAREA
  WMEMNAME 100
  M1 8  1  M2 8  9  M3 8 17  M4 8 25  M5  8 33
  M6 8 41  M7 8 49  M8 8 57  M9 8 65  M10 8 73

DYL: CALL DYLPDS USING OPTCODE DIR MEMRECORD
  IF STATUSCODE EQ 'E' GOTO ERR ENDIF
  IF STATUSCODE EQ X'FF' GOTO EOF ENDIF
  MOVE MEMNAME TO WMEMNAME (INW)
  IF INW EQ 80 GOTO PRT ENDIF
INC: INW=INW + 8 GOTO DYL
PRT:
  PERFORM LISTDET TO LISTEND
  MOVE SPACES TO WMEMNAME
DUM:
CLR: MOVE 00 TO INW GOTO DYL
ERR: MOVE 'ERROR?' TO DYLPRTCOMM
  PRINT MEMRECORD
```

Figure 110 Example 2 VISION:Results Code (Page 1 of 2)

```
EOF: PERFORM PRT TO DUM
EOJ: STOP
LISTDET:
  LIST M1 (MEMBER'NAME)  M2 (MEMBER'NAME)  M3 (MEMBER'NAME)
    M4 (MEMBER'NAME)  M5 (MEMBER'NAME)  M6 (MEMBER'NAME)
    M7 (MEMBER'NAME)  M8 (MEMBER'NAME)  M9 (MEMBER'NAME)  M10 (MEMBER'NAME)
LISTEND:
T1 'COMPUTER ASSOCIATES   TEST DYLPDS—EXAMPLE 2'
T2 'VISION:RESULTS SUPPORT UTILITY—-DYLPDS'
T3 'CA MACLIB' WITH 2 AFTER
```

Figure 110 Example 2 VISION:Results Code (Page 2 of 2)

```
COMPUTER ASSOCIATES  TEST DYLPDS—EXAMPLE 2
                                VISION:RESULTS SUPPORT UTILITY—DYLPDS
                                          CA MACLIB

MEMBER     MEMBER     MEMBER     MEMBER     MEMBER     MEMBER     MEMBER     MEMBER     MEMBER     MEMBER
 NAME       NAME       NAME       NAME       NAME       NAME       NAME       NAME       NAME       NAME

AADRV00    ADDRV00    AFDRV00    AIDRV00    AIPSW00    ALCNF00    ALDRV00    ALFQA00    APERR00    APSMP00
ARABD00    ARCPE00    ARCP00     ARCP00D    ARCP100    ARDBLWD    ADDRV05    ARENDOOD   ARERR00    ARFP00
ARI00      ARID00D    ARPHTAB    ARPH00D    ARPUB00D   ARSPY00    ARSPY00D   ARSR000    ASEND00    ASMNONE
ASPSNCOM   ASPSN00     ASRNG00    ASRPS00    AVAGE00    AVCNF00    AVCNW00    AVDRV00    AVFUG00    AVFRQ00
AVRPS00    AVRRN00    AVSR00     AVSBI00    AVSRN00    AVSRT00    AVSSM00    AVTMG00    AWCNR00    AWCTL00
AWDRV00    AWDYL00    AWID000    AZDRV00    AZSPY00    DAENATF    DAENATI    DAENCOM    DAENDAT    DAENDDV
DAENDFQ    DAENDHS    DAENDMY    DAENDPS    DAENDSP    DAENDST    DAENDVR    DAENDWS    DAENPKF    DAENPSN
DAENSRT    DAENSRT2   DAINDVR    DAINITZ    DAINLOAD   DAINRFLD   DAINSCB    DAINSCN    DAPSAG     DAPSCN
DAPSFQ     DAPSRAN    DAPSSP     DARIDBW    DARIDID    DARIDVR    DARIFPT    DARIMID    DARIPHT    DARIPRID
IJFVZZWZ   IJFVZZZZ   IJGFOZZZ   IJGVIZZZ   IJGXOZZZ   IJJFCBID   IVCNCB     IVFQCB     NEGIRY     TRYTIME
XXXX
```

Figure 111 Example 2 Output

Example 3 List EXEC Statements of Selected Procedure Library Members — MSEL Function

In this example, you access a procedure library (SYS1.PROCLIB) and interrogate four selected members for the purpose of extracting and printing their respective execute statements.

The MSEL (member select) function uses the LET option (L in position 5 of input parameter) to continue processing even if a member, other than the last, is not found in the directory search.

Step by Step

This example is similar to examples 1 and 2 with some exceptions. Only those lines of procedure logic that are different from the previous examples are discussed in this example.

1. Because you are searching for EXEC statements exclusively, you must examine each member record byte by byte. You can use the VISION:Results indexing feature to accomplish this.

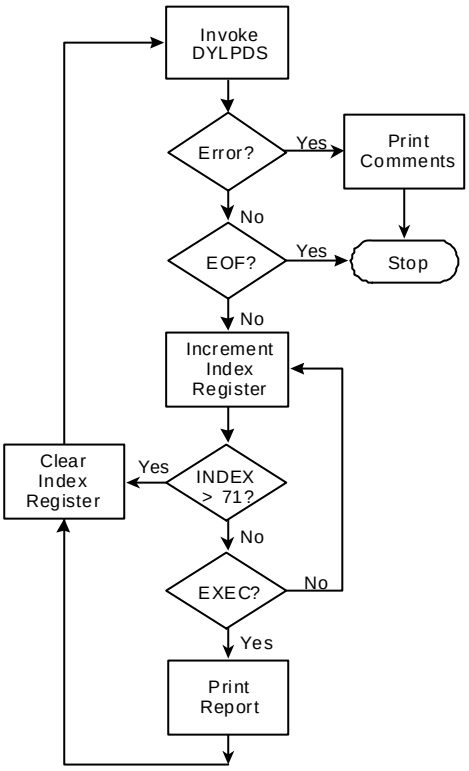

Figure 112 Example 3 Procedure Logic Flowchart

```
INC: INW=INW+1
IND: IF INW GT 71 GOTO CLR ENDIF
```

The INC statement adds a positive value of 1 to index INW.

2. The IND statement checks to see if INW has gone past column 71 (the last valid JCL column), in a search for EXEC. If it has gone beyond column 71, the INW counter is cleared to binary zeros and the next member record is read.

```
CLR: MOVE 00 TO INW GOTO DYL
```

3. If INW is 71 or less, 6 bytes in MEMRECORD at location (MEMEXEC) indexed by INW, is compared to the literal EXEC. If a match is not found, a branch to INC is taken to increment the index counter. If a match is found, the execute statement is printed, the index counter is reset to zeros, and the next member record is read.

```
IF MEMEXEC EQ ' EXEC ' NEXT ELSE GOTO INC ENDIF
PERFORM LISTDET TO LISTEND
CLR: MOVE 00 TO INW  GOTO  PDS
```

```
FILE SYSIN CARDS DUMMY
WORKAREA
  OPTCODE 52  FUNCTION 4 1 VALUE 'MSEL'  LETOPT 1 5 VALUE 'L'
  ESTNO 1 6 VALUE 1  MEMBER  30  9 VALUE 'ACBGEN,ASMFC,ASMFCL,ASMFCLG'
  DSNAME 44 1  VOLID 6 45  RECLEN 2 51 BI
WORKAREA
  DIR 75  STATUSCODE 1 1  DIRENTRY 74 2  MEMNAME 8  2
WORKAREA
  MEMRECORD 80  ERRORMSG  1  1  MEMEXEC   6  1

DYL: CALL DYLPDS USING OPTCODE DIR MEMRECORD
  IF STATUSCODE EQ 'E' GOTO ERR ENDIF
  IF STATUSCODE EQ X'FF' GOTO EOJ ENDIF
INC: INW=INW+1
IND: IF INW GT 71 GOTO CLR ENDIF
  IF MEMEXEC (INW) EQ ' EXEC ' NEXT ELSE GOTO INC ENDIF
  PERFORM LISTDET TO LISTEND
CLR: MOVE 00 TO INW GOTO DYL
ERR:  MOVE 'ERROR?'  TO  DYLPRTCOMM
  PRINT  MEMRECORD
EOJ: STOP

CONTROL MEMNAME

LISTDET:
  LIST  SUPPRESS MEMNAME (MEMBER NAME) MEMRECORD (CONTENTS OF MEMBER)
LISTEND:

ON CHANGE IN MEMNAME
  LIST TALLY (NUMBER OF EXECUTES)

ON FINAL
  LIST TALLY
T1 'COMPUTER ASSOCIATES  TEST DYLPDS—EXAMPLE 3'
T2 'VISION:RESULTS SUPPORT UTILITY—-DYLPDS'
T3 'SYS1.PROCLIB' WITH 2 AFTER
```

Figure 113 Example 3 VISION:Results Code

```
COMPUTER ASSOCIATES  TEST DYLPDS—EXAMPLE 3
                        VISION:RESULTS SUPPORT UTILITY—-DYLPDS
                                   SYS1.PROCLIB

MEMBER NAME             CONTENTS OF MEMBER                    NUMBER OF EXECUTES

 ASMFC      //ASM    EXEC  PGM=IEUASM,PARM=NODECK,REGION=70K
                                                                              1
 ASMFCL     //ASM    EXEC  PGM=IEUASM,PARM='NODECK,LOAD',REGION=70K
            //LKED   EXEC  PGM=IEWL,PARM=(XREF,LIST,NCAL),REGION=96K,
                                                                              2
 ASMFCLG    //ASM    EXEC  PGM=IEUASM,PARM='NODECK,LOAD',REGION=70K
            //LKED   EXEC  PGM=IEWL,PARM=(XREF,LET,LIST,NCAL),REGION=96K,
            //GO     EXEC  PGM=*.LKED.SYSLMOD,COND=((8,LT,ASM),(4,LT,LKED))
                                                                              3
                                                                              6
```

Figure 114 Example 3 Output

Example 4 Back Up a PDS to Tape Adding IEBUPDTE Control Statements Before Each Member — MALL Function

In this example, all members of a macro library are read from SYS1.MACLIB and written to tape. An IEBUPDTE ADD control statement is written before each member. This is done so that the library can be restored using the IBM IEBUPDTE utility.

The IEBUPDTE control statement is being printed as it is written to tape. This provides a complete list of the members of this library.

Step by Step

1. The first four statements respectively call DYLPDS, test for an end of file, test for an error condition, and check for the new member status indicator.

```
DYL: CALL DYLPDS USING OPTCODE DIR MEMRECORD
  IF STATUSCODE EQ X'FF' GOTO EOJ ENDIF
  IF STATUSCODE EQ 'E' GOTO ERR ENDIF
  IF STATUSCODE EQ 'N' GOTO IEB ENDIF
```

   The VISION:Results data areas utilized are:

Parameter	Data Area		Use
1	OPTCODE	To DYLPDS:	DYLPDS function code MALL.
		From DYLPDS:	Data set name, volume serial, logical record length.
2	DIRECTORY	To DYLPDS:	N/A.
		From DYLPDS:	Status indicator and directory records.
3	MEMRECORD	To DYLPDS:	N/A.
		From DYLPDS:	Member records or error messages.

2. If end-of-file or error conditions do exist, a branch to the appropriate statement is made. If a new member is not detected, the member record is moved to the output area (BUFFER1) and written to the VISION:Results output file, FILEJ, and the VISION:Results operating cycle is completed (by using ACCEPT).

3. If the new member indicator is on, the name of the new member is moved from MEMNAME to SELECTED and the work area ADDNAME for 20 is moved to BUFFER1.

```
WRT: MOVE MEMRECORD TO BUFFER1
     WRITE FILEJ ACCEPT
IEB: MOVE MEMNAME TO SELECTED
     MOVE ADDNAME LENGTH 20 TO BUFFER1
```

4. The next statement issues a WRITE immediate command. This places the IEBUPDTE control statement before the next new member.

```
WRITE FILEJ
```

5. The next statement causes the IEBUPDTE control statement to be printed.

```
PRINT BUFFER1
GOTO WRT
```

6. After the file print is performed, a branch to WRT enables the first record of the new member to also be written to tape.
7. The last three statements establish a return code of 502, print the error message, and set the end of job indicator to terminate processing.

```
ERR: MOVE 502 TO DYLRETURN
     PRINT MEMRECORD
EOJ: STOP
```

8. This program could be easily modified to select only certain members (MSEL) or to alter members as they are written.

```
FILE SYSIN CARDS DUMMY
FILE FILEJ FB 80 80 OUTPUT FROM FILEJ
  BUFFER1 80 VALUE '  ' REINIT

WORKAREA
  OPTCODE 52  FUNCTION 4 1 VALUE 'MALL'
WORKAREA
  DIR 75  STATUSCODE 1 1  DIRENTRY 74 2  MEMNAME 8 2
WORKAREA
  MEMRECORD 80  ERRORMSG 1 1  MEMEXEC 6 1
WORKAREA
  ADDNAME 12 VALUE './ ADD NAME='  SELECTED 8 VALUE ' '

DYL: CALL DYLPDS USING OPTCODE DIR MEMRECORD
  IF STATUSCODE EQ 'E' GOTO ERR ENDIF
  IF STATUSCODE EQ X'FF' GOTO EOJ ENDIF
  IF STATUSCODE EQ 'N' GOTO IEB ENDIF

WRT: MOVE MEMRECORD TO BUFFER1
  WRITE FILEJ ACCEPT

IEB: MOVE MEMNAME TO SELECTED
  MOVE ADDNAME LENGTH 20 TO BUFFER1
  WRITE FILEJ
  PRINT BUFFER1
  GOTO WRT

ERR: MOVE 502 TO DYLRETURN
  PRINT MEMRECORD
EOJ: STOP
```

Figure 115 Example 4 VISION:Results Code

```
./ ADD NAME=AUDCOD
./ ADD NAME=COBUCL
./ ADD NAME=COBUCLG
./ ADD NAME=COD
./ ADD NAME=CODBETA
./ ADD NAME=CODOSTST
./ ADD NAME=CODPRINT
./ ADD NAME=CODPROD
./ ADD NAME=CODTEST
./ ADD NAME=CONDENSE
./ ADD NAME=CRDDIR
./ ADD NAME=CRDLDDIR
./ ADD NAME=DOSASMG
./ ADD NAME=DOSASMG2
./ ADD NAME=ELORAC
./ ADD NAME=GBLTEST
./ ADD NAME=LOWUP
./ ADD NAME=LOWUP2
./ ADD NAME=LOWUP3
./ ADD NAME=LOWUP4
./ ADD NAME=LOWUP5
./ ADD NAME=LSTDIR
./ ADD NAME=LSTDIR2
```

Figure 116 Example 4 Output

Example 5 List Attributes of All Load Modules in a PDS with No Printing of Member Records — DALL Function

This example uses the DALL function of DYLPDS to list the member names and attributes of all members in the load library Z01.DY.PRODLB.

The DALL function is provided to DYLPDS in the first four bytes of the work area FUNCTION.

```
FILE SYSIN CARDS DUMMY

WORKAREA
  OPTCODE 52  FUNCTION 4 1 VALUE 'DALL'  DSNAME 44 1
  VOLID 6 45  RECLEN 2 51 BI
WORKAREA
  DIR 75  STATUSCODE 1 1  DIRENTRY 74 2  MEMNAME 8 2
  INDICATOR 1 13 BI Z  SIZE 3 24 BI E  ATTR1 1 22 BI
  ATTR2 1 23 BI
WORKAREA
  MEMRECORD 100  ERRORMSG 1 1  MEMEXEC 6 1  RENT 1 81 NU Z
  REUS 1 82 NU Z  OVLY 1 83 NU Z  TESTRAN 1 84 NU Z
  OL 1 85 NU Z  SCTR 1 86 NU Z  EXEC 1 87 NU Z
  TXT 1 88 NU Z  NOTDC 1 89 NU Z  ORG 1 90 NU Z
  EPZERO 1 91 NU Z  NORLD 1 92 NU Z  NOTEDIT  1  93 NU Z
  SYMS 1 94 NU Z  FLEVEL 1 95 NU Z  REFR 1 96 NU Z

DYL: CALL DYLPDS USING OPTCODE DIR MEMRECORD
  IF STATUSCODE EQ 'E' GOTO ERR ENDIF
  IF STATUSCODE EQ X'FF' GOTO EOJ ENDIF
PRT: CALL DYLBZON USING ATTR1 RENT
  CALL DYLBZON USING ATTR2 NOTDC
  PERFORM LISTDET TO LISTEND
  GOTO DYL
ERR: MOVE 'ERROR' TO DYLPRTCOMM
  PRINT MEMRECORD
EOJ: STOP

CONTROL MEMNAME

LISTDET:
LIST SUPPRESS MEMNAME (MEMBER'NAME) INDICATOR RENT REUS
  OVLY TESTRAN OL SCTR  EXEC  TXT  NOTDC (NOT'DC)
  ORG (ZERO'ORG) EPZERO (EP'ZERO)  NORLD (NO RLD)
  NOTEDIT (NOT'EDIT)  SYMS  FLEVEL REFR  SIZE
LISTEND:
ON CHANGE IN MEMNAME
  MOVE 99 TO DYLLINE
T1 'COMPUTER ASSOCIATES   TEST DYLPDS—EXAMPLE 5' WITH 2 AFTER
```

Figure 117 Example 5 VISION:Results Code

```
COMPUTER ASSOCIATES TEST DYLPDS—EXAMPLE 5

MEMBER INDICATOR RENT REUS OVLY TESTRAN OL SCTR EXEC TXT NOT ZERO EP  NO RLD NOT SYSMS FLEVEL REFR SIZE
 NAME                                          DC  ORG ZERO       EDIT

ALDRV00   43                                1      1   1   1                  1        4,096
APERR00   43                                1      1   1   1                  1       19,152
ARABD00   43                                1   1  1   1   1     1            1          744
ARCPE00   43                                1      1   1   1     1            1        6,280
ARDRV05   43                                1      1   1   1                  1        2,544
ARERR00   43                                1      1   1   1                  1          704
ARFP00    43                                1   1  1   1   1     1            1        1,648
ARIO00    43                                1      1   1   1                  1        1,912
ARIO00D   43                                1      1   1   1                  1        3,512
ARPUB00D  43                                1   1  1   1   1     1            1        1,200
ARSPY00   43                                1      1   1   1                  1          608
ARSPY00D  43                                1   1  1   1   1     1            1          720
AWIO00    43                                1      1   1   1                  1        1,024
AWIO00D   43                                1      1   1   1                  1        4,208
```

Figure 118 Example 5 Output

Example 6 Scan and Select Members Using a Prefix — DALL Function with PASS (P) Option

This example uses the DALL function of DYLPDS to scan a directory looking for all members whose first five characters are PDSEX. When a member is found, the PASS option (P) is turned-on to return the source code to the calling program.

```
FILE SYSIN CARDS DUMMY

WORKAREA
     OPTCODE  52  FUNCTION  4  1  VALUE  'DALL'
                  DSNAME  44  1
                  VOLID  6  45  RECLEN  2  51  BI
WORKAREA
     DIR 75  STATUSCODE 1 1  DIRENTRY 74  2  MEMNAME  8  2
             (MEMBER'NAME)  REDEFINE MEMNAME
             MEMPART  5  FILLER  3
WORKAREA
     MEMRECORD  80  (SOURCE FROM SELECTED MEMBERS)

RDPDS:   CALL DYLPDS USING OPTCODE DIR MEMRECORD
       IF STATUSCODE  EQ  'E'  GOTO  ERR   ENDIF
       IF STATUSCODE  EQ X'FF'  GOTO  EOF   ENDIF
       IF MEMPART EQ 'PDSEX' NEXT ELSE REJECT  ENDIF
       IF STATUSCODE EQ 'N' NEXT ELSE GOTO NOW  ENDIF
       MOVE  'P'  TO  STATUSCODE
NOW:
       PERFORM  LISTDET  TO  LISTEND
       MOVE  SPACES  TO  MEMRECORD
           GOTO  RDPDS

ERR:   MOVE  'ERROR?'  TO  DYLPRTCOMM
       HEXPRINT  OPTCODE
       HEXPRINT  DIRECTORY
EOF:   STOP

LISTDET:
       LIST  SUPPRESS  MEMNAME  MEMRECORD
LISTEND:

T1   'COMPUTER ASSOCIATES    TEST DYLPDS—EXAMPLE 6'
T2   'VISION:RESULTS SUPPORT UTILITY—-DYLPDS'
T3   ' ' WITH 2 AFTER
T3+1 DSNAME
```

Figure 119 Example 6: DALL Function

```
COMPUTER ASSOCIATES  TEST DYLPDS—EXAMPLE 6
                  VISION:RESULTS SUPPORT UTILITY—-DYLPDS

CN040950.DYL.PUB.TESTPROG
                                      SOURCE FROM SELECTED MEMBERS
MEMBER
 NAME

PDSEX001
           FILE SYSIN CARDS DUMMY

           WORKAREA
                OPTCODE 52  FUNCTION 4  1  VALUE 'MSEL'  LETOPT  1  5
                             VALUE  'L'  MEMBER  18  9  VALUE
                             'GETMAIN,CLOSE,OPEN'  DSNAME  44  1
                             VOLID  6  45  RECLEN  2  51  BI
           WORKAREA
                DIR 75  STATUSCODE 1  1  DIRENTRY 74 2
                        MEMNAME 8  2
           WORKAREA
                MEMRECORD  80  ERRORMSG  1  1

           DYL:  CALL  DYLPDS USING OPCODE DIR MEMRECORD
```

Figure 120 Example 6 Output (Page 1 of 2)

```
                IF STATUSCODE EQ 'E' GOTO ERR  ENDIF
                IF STATUSCODE EQ X'FF' GOTO EOJ  ENDIF
                PERFORM LISTDET TO LISTEND
                    GOTO DYL
           ERR:  MOVE 'ERROR?' TO DYLPRTCOMM
           *      PRINT MEMRECORD
           EOJ:   STOP
           CONTROL MEMNAME
           LISTDET:
                 LIST SUPPRESS MEMNAME (MEMBER NAME) MEMRECORD
                      (CONTENTS OF MEMBER)
           LISTEND:
           T1   'COMPUTER ASSOCIATES  TEST DYLPDS—EXAMPLE 1'
                          WITH 2 AFTER
           FIN
PDSEX002
           FILE SYSIN CARDS DUMMY
           WORKAREA
                OPTCODE 52  FUNCTION 4  1  VALUE 'DALL'
                            DSNAME 44  1
                            VOLID 6  45  RECLEN 2  51  BI
```

Figure 120 Example 6 Output (Page 2 of 2)

PDS Directory Format

The following table describes the PDS directory format returned in parameter 2, bytes 2-75, from DYLPDS.

Byte	Length (in Bytes)	Description
2	8	Member or alias name.
10	3	Relative address FAll addresses are relative to the first block of the data set. TT = Relative track from beginning of data set. R = Block number of that track. TTR of first block of member
13	1	Indicators: ■ Bit 0=1 — Name in bytes 2-9 is an alias ■ Bits 1=2 — Number of TTRs in bytes 14-35 ■ Bits 3=7 — Length of user data field in half words
14	3	TTR of first block of text.
17	1	Not used.
18	3	TTR of node list or scatter table for modules in scatter load or overlay structure only.
21	1	Number of entries in node list for modules in overlay structure only.

Byte	Length (in Bytes)	Description
22	2	Attributes of load module. **Byte 1** Bit 0=1 Reenterable. Bit 1=1 Reusable. Bit 2=1 Overlay structure. Bit 3=1 Testran. Bit 4=1 Loadable only. Bit 5=1 Scatter format. Bit 6=1 Executable. Bit 7=1 No relocation dictionary one block of text. **Byte 2** Bit 0=1 F level linkage editor only. Bit 0=0 All levels of linkage editor. Bit 1=1 Origin of first block of text=0. Bit 1=0 Origin of first block of text=0. Bit 2=1 EPA is zero. Bit 3=1 No relocation dictionary items. Bit 4=1 Module cannot be processed by linkage editor. Bit 5=1 Module contains testran cards. Bit 6=1 Module created by level F linkage editor.
24	3	Contiguous memory requirements.
27	2	Block length of first text block.
29	3	EPA — Entry point address.
32	3	Linkage editor assigned origin of first test block.

If module has an alias name and is reusable or reenterable:

Byte	Length (in Bytes)	Description
35	3	Entry point.
38	8	Member name when bytes 2-9 contain an alias name.

If load module is in scatter format:

Byte	Length (in Bytes)	Description
35	2	Size of scatter list.
37	2	Bytes in translation table.
39	2	External symbol directory identification of the CSECT to which first block of text belongs.
41	2	ESDID of CSECT containing entry point.

Scatter format with reenterable and reusable attributes:

Byte	Length (in Bytes)	Description
43	3	Entry point.
46	8	Member name if name in bytes 2-9 is an alias.

Chapter 11: VISION:Table

VISION:Results provides the same facility by using the TABLE, STORE, RETRIEVE, TSORT, TSIZE, SEARCH, BINSEARCH, and DELETE statements and commands. The discussion in this chapter is for compatibility with programs still using the VISION:Table product.

VISION:Table is an optional product designed to give extensive table handling capability to z/OS and VSE users. The following fully automated functions are supported for up to 16 concurrent tables:

- Table load and reload and delete.
- Sequential retrieval (any starting point).
- Random retrieval.
- Serial search based on any key.
  - Starting from the beginning.
  - Starting at any entry number in the table.
- Binary search based on any key.

Typical applications for VISION:Table might be:

- Load a description file into a table and retrieve entries by key.
- Build and maintain a bank of accumulators having a one-to-one relationship with some variable such as department number or state code.
- Buffer input records and retrieve later for processing.

VISION:Table is provided as a module called DYLTABLE which is a 6K routine invoked using a CALL statement in procedure logic. You supply an area between 1024 and 262136 bytes in size, and a few pieces of information such as entry length and the location of the data to be stored. After the table is loaded, you need only supply the entry number at which to start retrieval or the key length and location on which to search.

Loading a Table

If you want to store entries (fixed length only) in a table, the following procedure should be used:

- Allocate an area to hold the table.

```
WORKAREA    TBLAREA    10240
```

The area allocated must be larger than the number of bytes needed to hold the table entries; this is because of control information used by DYLTABLE. Information regarding this can be found in Considerations for Use on page 269.

- Set up the following table control block in a separate work area. (Do not use the same area that is to hold the table.)

```
WORKAREA
  OPCODE    4    1       VALUE  'PUT'
  TABLENUM  1    5   BI  VALUE  nn
  ENTRYLEN  2    6   BI  VALUE  nnnnn
  ENTRYNUM  2    8   BI  VALUE  1
  KEYLENGTH 1   10   BI  VALUE  n
  KEYLOCN   1   11   BI  VALUE  n
  TABLESZE  4   12   BI  VALUE  nnnnnn
  RETCODE   1   16   BI  VALUE  0
  MESSAGE  32   17       VALUE  ' '
```

Figure 121 Table Control Block Example

where:

OPCODE specifies the function to be done by DYLTABLE. It can have the value of PUT, GET, REPL, FIND, FINR, SRCH, DELT.

TABLENUM specifies the table number. It can have a value in the range of 0 to 15.

ENTRYLEN specifies the length of each entry. It can have a value in the range of 1 to 32767.

ENTRYNUM specifies the entry number to be stored or retrieved. It can have a value in the range of 1 to 32767.

KEYLENGTH specifies the length of the key in the entry. It can have a value in the range of 1 to 255. A value is required here only for FIND, FINR, or SRCH operations.

KEYLOCN specifies the location of the key in the table entry. It can have a value in the range of 1 to 255. A value is required only for FIND, FINR, or SRCH operations.

TABLESZE specifies the size of the table area to be used by DYLTABLE and can have a value in the range of 1024 to 262136.

RETCODE holds the return code passed back by DYLTABLE to indicate the success of the request:

0 = Successful completion.

1 = End of table reached on a retrieval or entry number beyond final table entry.

2 = Entry not found (FIND, FINR, SRCH operations).

6 = Invalid entry length (PUT operation).

7 = Requested table number not found.

8 = Table area is full.

9 = Invalid or missing parameter or operation code.

MESSAGE holds an error description returned by DYLTABLE in the event of an unsuccessful operation. This message can be printed out by the user.

Code procedure logic statements to cause the entries to be loaded into a table. Typically, you would read a file containing the entries, and then start DYLTABLE through the CALL facility, passing it three parameters:

- The field name of the DYLTABLE control block.
- The field name of the table area.
- The field name of the entry to be stored.

Example 1 Create a Table of 40-byte Account Number or Account Name Entries

The first five bytes of the entry contain the account number, followed by a 35-byte account name. The entries to be stored in the table are on the second input file, FILEB. Both the DYLTABLE control block and table area are defined in a work area.

```
FILE   FILEA   FB   200   2000
  ACCTKEY      5     1
  ACCTNO       5     2
FILE   FILEB   FB   40    400
  FILEBAREA    40
  COUNTER      3     6     PD
WORKAREA
  TBLAREA   10240

WORKAREA
  OPCODE    4   1       VALUE  'PUT'
  TABLENUM  1   5   BI  VALUE  0
  ENTRYLEN  2   6   BI  VALUE  40
  ENTRYNUM  2   8   BI  VALUE  1
  KEYLENGTH 1  10   BI  VALUE  5
  KEYLOCN   1  11   BI  VALUE  1
  TABLESZE  4  12   BI  VALUE  10240
  RETCODE   1  16   BI  VALUE  0
  MESSAGE  32  17       VALUE  ' '
PHASE1:
ON ONE
  PERFORM LOADTB1 TO LOADTB9
ENDONE
PHASE2: READ FILEA
LOADTB1: READ FILEB
  IF FILEB EQ X'FF' GOTO LOADTB9 ENDIF
  CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA
  IF RETCODE GT 0 GOTO ERRTN ELSE GOTO LOADTB1 ENDIF
LOADTB9:
```

Figure 122 Example Table

A one-time-only (ON ONE to ENDONE) PERFORM command statement is issued in the procedure logic to read FILEB and give entries to DYLTABLE until FILEB is exhausted. Then regular processing begins.

Retrieving an Entry by Key

If you want to retrieve a specific entry by key, starting at the beginning of the table, do the following in your procedure statements:

- Modify the same control block you used for loading the table:
  - Move a FIND to OPCODE in the control block.
  - If you have not set them up initially in the WORKAREA field definitions, move the correct values to the KEYLENGTH and KEYLOCN fields.
- Start DYLTABLE, passing it four parameters:
  - The field name of the DYLTABLE control block.
  - The field name of the table area.
  - The field name of the area where the found entry is to be placed.
  - The field name of the search key.
- Optionally, check the return code for a found or not found condition:
  - 0 = Entry found.
  - 2 = Entry not found.

Example 2 Retrieve an Account Number or Account Name Entry by Key

Using the table created in the previous example, retrieve an Account Number or Account Name entry based on a 5-byte Account Number field from the input master file FILEA. The 5-byte Account Number starts in the second byte of the master file. If the entry is found, DYLTABLE moves it to FILEBAREA, as specified in the parameter list.

```
PHASE2:  READ FILEA
  ON ONE
    MOVE 5 TO KEYLENGTH
    MOVE  2 TO KEYLOCN
    MOVE 'FIND' TO OPCODE
  ENDONE
  CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA ACCTNO
  IF RETCODE EQ 0 NEXT ELSE GOTO NOTFOUND ENDIF
```

Figure 123 Actions to Example Table on page 261

Retrieving an Entry by Key—Using a Specified Number

If you want to locate a specific entry by key, but you do not want to start at the beginning of the table, you can use a specified number. This might be the case if you are retrieving multiple occurrences of entries with the same key or if you know that a particular entry has to be at or after a certain entry number.

Code procedure statements to do the following:

- Modify the same control block you used for creation of the table:
  - Move a FINR (Find Resume) to OPCODE in the control block.

  - If you have not set them up initially in the WORKAREA field definitions, move the appropriate values to the KEYLENGTH and KEYLOCN fields.
  - Move the appropriate entry number at which to start the search to the ENTRYNUM field.
- Start DYLTABLE, passing it four parameters:
  - The field name of the control block.
  - The field name of the table area.
  - The field name of the area where the found entry is to be placed.
  - The field name of the key for which you want DYLTABLE to search.
- Optionally, check the return code in the control block for a found, not found, or beyond range condition:
  - 0 = Found.
  - 1 = Beyond end of table.
  - 2 = Not found.

Example 3 Using Account Number Keys

This example uses a table where the entries contain an account number key that is in ascending sequence. The input master file is also in account number sequence. This example uses the FINR op code to search the table. This search does not have to start at the beginning each time, because the account number is in sequence so you can start a search from the next entry number following the last account number found, assuming the key length and key location have been set previously in the table control block.

```
ON ONE
  MOVE 0 TO ENTRYNUM
  MOVE 'FINR' TO OPCODE
ENDONE
ENTRYNUM = ENTRYNUM + 1
CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA ACCTNO
IF RETCODE EQ 0 NEXT ELSE GOTO NOTFOUND ENDIF
```

Figure 124 Example Table: Account Number Key in Ascending Sequence

Note: For SKIP SEQUENTIAL processing starting at a particular key, you should use a FIND followed by GET until the end of the table.

Retrieving Each Entry

If you want to retrieve each entry in a table, one after the other, code the following in your procedure statements:

- Modify the same control block you used for the loading of the table:
  - Move GET to OPCODE in the control block.
  - If this is the first entry to be retrieved, move 1 to ENTRYNUM in the control block; otherwise, leave the entry number alone. DYLTABLE increments the entry number automatically as it retrieves the entry, so subsequent GET functions retrieve the next entry.
- Start DYLTABLE, passing it three parameters:
  - The field name of the control block.
  - The field name of the table area.
  - The field name of the area where the found entry is to be placed.
- Optionally, check the return code in the control block for an end-of-table condition (RETCODE = 1).

Example 4 Retrieve Each Entry from the Account Number or Name Table

Begin with entry number 1 and have each entry returned to FILEBAREA.

Note that the moving of the 1 to the ENTRYNUM field and the GET to the OPCODE field are one-time-only statements.

```
ON ONE
  MOVE  1    TO ENTRYNUM
  MOVE 'GET' TO OPCODE
ENDONE
CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA
IF RETCODE EQ 1 GOTO EOJ ENDIF
```

Figure 125 Retrieving Entries Example

Retrieving an Entry by Entry Number

If you want to retrieve a particular entry by entry number in a table, do the following in procedure statements:

- Modify the same control block you used for the loading of the table:
  - Move GET to OPCODE in the control block.
  - Move the entry number of the entry to be retrieved to the ENTRYNUM field.

- Start DYLTABLE, passing it three parameters:
  - The field name of the control block.
  - The field name of the table area.
  - The field name of the area where the found entry is to be placed.
- Optionally, check the return code in the control block for successful completion:
  - 0 = Record retrieved.
  - 1 = End-of-table.

Example 5 Retrieve an Entry by Entry Number

Retrieve the eighth entry in the Account Number and Name table.

```
ON ONE
  MOVE 'GET' TO OPCODE
ENDONE
  MOVE 8 TO ENTRYNUM
  CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA
```

Figure 126 Retrieving Entries Example

Retrieving an Entry by Key—Using a Binary Search

If you have a table of 20 entries or more that are in **ascending sequence by key**, you could perform a binary search. For such a table, a binary search is much faster than a serial search. Do the following in your procedure statements:

- Modify the same control block you used for the loading of the table:
  - Move SRCH to OPCODE in the control block.
  - If you have not set them up initially in the WORKAREA field definitions, move the correct values to the KEYLENGTH and KEYLOCN fields.
- Start DYLTABLE, passing it four parameters:
  - The field name of the DYLTABLE control block.
  - The field name of the table area.
  - The field name of the area where the found entry is to be placed.
  - The field name of the key in the record (not entry) to be searched for.
- Optionally, check the return code in the control block for a found or not found condition:
  - 0 = Found.
  - 2 = Not found.

Example 6 Retrieve a Particular Entry by Account Number Key Using Binary Search

Retrieve a particular entry by Account Number key using the binary search technique. Assume the entry key length (5) and key location (1) values have already been set in the control block. The key in the input record is in location 2.

```
ON ONE
  MOVE 'SRCH' TO OPCODE
ENDONE
CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA ACCTNO
IF RETCODE GT 0 GOTO NOTFOUND ENDIF
```

Figure 127 Retrieving Entries Example

Each invocation of the SRCH function either does or does not find the record. This function need not be executed repeatedly to find that record. One CALL does the entire search.

Replacing an Entry in a Table

If you want to update a particular entry in a table, do the following in procedure statements:

- Modify the same control block you used for loading the table:
  - Move REPL to the OPCODE field.
  - Move the entry number of the entry you want replaced to the ENTRYNUM field, if it does not already contain it.
- Start DYLTABLE, passing it three parameters:
  - The field name of the DYLTABLE control block.
  - The field name of the table area.
  - The field name of the entry that is to replace the old entry.
- Optionally, check the return code field in the control block for successful completion:
  - 0 = Successful completion.
  - 1 = Entry number greater than number of entries in table.

Example 7 Replacing Entries

Each entry in a table contains a 3-byte packed counter starting in location 6 (COUNTER). A particular entry's counter field is incremented by one if its key (location 1) matches a key contained in the first five bytes of the input record (ACCTKEY). In order to update the entry, it is first retrieved by using the FIND operation code; then it is replaced after the counter has been incremented.

```
READ FILEA
MOVE 'FIND' TO OPCODE
CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA ACCTKEY
COUNTER = COUNTER + 1
MOVE 'REPL' TO OPCODE
CALL DYLTABLE 6K USING OPCODE TBLAREA FILEBAREA
```

Figure 128 Replacing Entries Example

Deleting a Table

You can delete a table in order to rebuild the table or to free up space in the table work area. To delete a table, do the following in procedure statements:

- Modify the same control block you used for the loading of the table:
  - Move a DELT to the OPCODE field of the DYLTABLE control block.
- Start DYLTABLE, passing it three parameters:
  - The field name of the DYLTABLE control block.
  - The field name of the table area.
  - Any valid subparameter to satisfy DYLTABLE's parameter requirement.

Example 8 Deleting a Table

Delete table number 1 in order to free up the space in the table area. Assume the table number is already in the table number field of the control block.

```
MOVE 'DELT' TO OPCODE
CALL DYLTABLE 6K USING OPCODE TBLAREA OPCODE
```

Figure 129 Delete Table Example

Error Messages

DYLTABLE indicates the result of each request in the return code (RETCODE) field of the control block area. It is advisable to check this return code following each CALL to DYLTABLE. Certain codes deserve special attention:

Error: 06 – Invalid Entry Length.
Cause: On a PUT (load) operation, the ENTRYLEN field in the control block contains a value less than 1 or greater than 32767.
Result: The PUT is not performed and DYLTABLE returns a message in the MESSAGE field of the control block, along with a return code of 6 in the RETCODE field. If you ignore the error and repeat the operation, the program continues returning a 6.
Action: Correct the entry length value and rerun the program.

Error: 07 – Requested Table Not Found.
Cause: A retrieve, replace, or delete operation is being specified for a table that does not exist.
Result: The requested operation is not performed and DYLTABLE returns a message in the MESSAGE field of the control block, along with a return code of 7 in the RETCODE field. If you ignore the error and repeat the operation, the program continues returning a 7.
Action: Provide the correct table number and rerun the program.

Error: 08 – Tabling Area is Full.
Cause: A PUT operation has been requested but there is insufficient room in the table area.
Result: The entry is not stored and DYLTABLE returns a message in the MESSAGE field of the control block, along with a return code of 8 in the RETCODE field. If you ignore the error and attempt to store another entry, DYLTABLE abends with a data exception. The VISION:Results error analysis prints an indicative error message in the section that lists the data fields involved in the data exception.
Action: Allocate additional work space in the appropriate work area or supply additional work areas. This increased allocation should be reflected in the TABLESZE field of the DYLTABLE control block. Rerun the program.

Error: 09 – Invalid or Missing Parameter or Opcode.
Cause: One or more of the required parameters for a given operation code, or the operation code itself, is either missing or invalid.
Result: DYLTABLE abends with a data exception. The VISION:Results error analysis prints an indicative message in the section that lists the data fields involved in the data exception.
Action: Check the CALL statements starting DYLTABLE and ensure that the correct number of parameters were passed and that the parameters specified the correct areas for the operation requested. Also, check the DYLTABLE control block to make sure that the correct values and defaults were present when the CALL to DYLTABLE was executed.

Considerations for Use

Determining the Proper Size of the Table Area

The size of the table work area (TWA) is determined chiefly by the amount of data you want to store. However, additional data required for table management is also placed in the table area by DYLTABLE, so the allocation must be greater than the number of bytes required for user entries. At least 1024 bytes of work area must be allocated. The following figure is provided as a guide to deciding how large to make the table area.

TWA Size	Max. Data Capacity
1024	510
2048	1530
4096	3570
8192	7650
16384	15810
32256	31620
65536	64260
131072	129030
262144	258570

Allocating a Large Table Work Area

A large table area (greater than 32768) can be obtained by allocating two or more adjacent work areas using WORKAREA commands. These areas are contiguous in memory.

```
WORKAREA
  TBLAREA1  32767
WORKAREA
  TBLAREA2  32767
```

Figure 130 Work Area Example

Multiple Tables

It is more convenient to find a table, when working with more than one table in a program, if you use a separate DYLTABLE control block for each table. You must, however, always pass DYLTABLE the same table work area. For example, if you are building Table 1 and Table 2, you pass the same TBLAREA to DYLTABLE.

Do not attempt to access entries in the table area without going through DYLTABLE.

Chapter 12: Starting VISION:Report

From your VISION:Results program, you can start stand-alone VISION:Report applications that access IMS or CA-IDMS/DB data bases. The following pages contain three examples of JCL statements and VISION:Report programs that illustrate how to accomplish this. While each example differs substantially, there are two items that must be included if VISION:Results is to successfully start VISION:Report. First, you must code the SYS004 statement in the JCL in exactly the format shown in the examples. Second, you must begin your VISION:Report program with the VISION:Results statement OPTION QUIKJOB.

The first example is the basic z/OS JCL used for starting VISION:Report from VISION:Results. The second example is the z/OS JCL for starting VISION:Report accessing CA-IDMS/DB, and the third example is the z/OS JCL for starting VISION:Report accessing IMS. Each example consists of two parts: the JCL statements and the VISION:Report program. The VSE JCL required for these jobs is the equivalent of the z/OS JCL.

There are certain aspects common to all three examples:

- The size of the REGION in the STEP01 statement is user-defined because it depends on the size of the program and what other applications it is accessing.
- Wherever the word user appears in lowercase in the JCL statements, a user-defined file name (if applicable) should be entered.

When the VISION:Report program contains access to CA-IDMS/DB data bases, the DYLQIDMS program should be invoked on the z/OS JCL EXEC statement. When the VISION:Report program contains access to IMS data bases, the DYLQKIMS program should be started for the same purpose. It should be noted that the installation procedures and the use of these two programs are the same as for the VISION:Report programs QUIKIDMS and QUIKIMS. (For more information, see the *Advantage VISION:Results for z/OS Reference Guide.*) See the examples – STEP01 statements – in VISION:Report z/OS JCL with CA-IDMS/DB on page 273 and VISION:Report z/OS JCL with IMS on page 276.)

To run VISION:Report from a VISION:Results program, you must be using VISION:Results Release 3.0 or later.

VISION:Report JCL

z/OS

```
                    //JOB  (your standard job statement)
                    //STEP01  EXEC PGM=DYL280,REGION=500K

VISION:Results      //STEPLIB   DD DSN=your.RESULTS.LOADLIB,
Release 3.0
or later
load library        //             DISP=SHR

VISION:Report       //          DD DSN=your.QUIKJOB.MVS.LOADLIB,
load library        //             DISP=SHR

                    //SYSPRINT  DD SYSOUT=*
                    //SYS280R   DD SYSOUT=*
                    //SYSOUT    DD SYSOUT=*

required            //SYS004    DD UNIT=SYSDA,SPACE=(TRK,(5))
format when
calling
VISION:Report

required            //SORTLIB   DD DSN=SYS1.SORTLIB,DISP=SHR
only if             //SORTWK01  DD UNIT=SYSDA,SPACE=(TRK,(20,20))
SORT is used        //SORTWK02  DD UNIT=SYSDA,SPACE=(TRK,(20,20))
                    //SORTWK03  DD UNIT=SYSDA,SPACE=(TRK,(20,20))

for VISION:Report    //SYSDET    DD *
card input only
(if applicable)

                     //SYSIN     DD *
```

Figure 131 z/OS Example

VSE

```
// JOB OPTIONQJ
VISION:Results      // DLBL DYL282,'RESULTS.SP60.LIBS'
PHASE LIB           // EXTENT ,DOS001

VISION:Report       // DLBL QIKCIL,'QUIKJOB.SP1306.CIL'
PHASE LIB           // EXTENT ,DOSLB1

                    // LIBDEF PHASE,SEARCH=(RESULTS.PHASE,QIKCIL.PHASE)
                    // ASSGN SYS004,DISK,VOL=DOSWRK,SHR
required            // DLBL IJSYS04,'==WORK1',0
                    // EXTENT SYS004.DOSWRK,1,0,1,10
                    // EXEC DYL280,SIZE=768K
                       OPTION QUIKJOB
                    .
                       .     VISION:Report program
                     .
                    /*
                    /&
```

Figure 132 VSE Example

Basic VISION:Report Program

OPTION QUIKJOB is a VISION:Results statement that must be present in order to start VISION:Report from VISION:Results.

The OPTION SEQCHK=NO statement is the VISION:Report option, and SEQCHK=NO is only one of the operands available. If no VISION:Report option statement is entered, the default values are used (see the *Advantage VISION:Results for z/OS Reference Guide*).

```
OPTION QUIKJOB

OPTION SEQCHK=NO


     1000   GET  DET ATEND EOJ
            MOVE  DET1-80 TO PRT1
            PRINT
            GO TO 1000
            END


ENTER INPUT DATA RECORDS FOR SYSDET FILE HERE.
```

Figure 133 OPTION QUIKJOB Example

VISION:Report z/OS JCL with CA-IDMS/DB

```
                //JOB  (your standard job statement)
                //STEP01  EXEC PGM=DYLQIDMS,REGION=2500K

VISION:Report   //STEPLIB   DD DSN=your.QUIKJOB.MVS.LOADLIB,
load library    //             DISP=SHR

VISION:Results  //          DD DSN=your.RESULTS.LOADLIB,
Release 3.0     //             DISP=SHR
or later
load library

CA-IDMS/DB      //          DD DSN=IDMS.PROD10.LOADLIB,
load library    //             DISP=SHR

user's          //          DD DSN=user.QUIK.LOAD,DISP=SHR
application
program library

user's copy     //SYSCOPY   DD DSN=user.DYLIDMS.MVS,
library         //             DISP=SHR

                //SYSPRINT  DD SYSOUT=*
                //SYS280R   DD SYSOUT=*
                //SYSOUT    DD SYSOUT=*

required        //SYS004    DD UNIT=SYSDA,SPACE=(TRK,(5))
format when
running
VISION:Report

required only   //SORTLIB   DD DSN=SYS1.SORTLIB,DISP=SHR
```

Figure 134 CA-IDMS/DB Example (Page 1 of 2)

```
if SORT is      //SORTWK01  DD UNIT=SYSDA,SPACE=(TRK,(20,20))
used            //SORTWK02  DD UNIT=SYSDA,SPACE=(TRK,(20,20))
                //SORTWK03  DD UNIT=SYSDA,SPACE=(TRK,(20,20))

                //SYSJRNL   DD DUMMY
                //J1JRNL    DD DUMMY
                //J2JRNL    DD DUMMY
                //J3JRNL    DD DUMMY
                //J4J4NL    DD DUMMY
```

Figure 134 CA-IDMS/DB Example (Page 2 of 2)

```
CA-IDMS/DB        //DICTDB    DD DSN=IDMS.PROD10.DICTDB,
dictionary        //             DISP=SHR

user's            //DYIDMSIN  DD DSN=user.QUIK.LOAD,DISP=SHR
VISION:Report
application library

CA-IDMS/DB        //ORGDEMO   DD DSN=IDMS.PROD10.ORGDEMO,
data base         //             DISP=SHR

for VISION:Report //SYSDET    DD *
card input only
(if applicable)
                  //SYSIN     DD *
```

Figure 135 CA-IDMS/DB Example

VISION:Report Program Calling CA-IDMS/DB Functions

OPTION QUIKJOB is a VISION:Results statement that must be present in order to start VISION:Report from VISION:Results.

The OPTION SEQCHK=NO statement is the VISION:Report option, and SEQCHK=NO is only one of the operands available. If no VISION:Report option statement is entered, the default values are used (see the *Advantage VISION:Results for z/OS Reference Guide*).

```
OPTION QUIKJOB

OPTION SEQCHK=NO

EQU BIND-RUN-UNIT       PCB155
EQU BIND-RN             PCB144
EQU READY-RETRIEVE      PCB133
EQU OBTN-1ST-RN-AN      PCB115
EQU OBTN-NXT-RP=AN      PCB107
EQU FINISH              PCB98
EQU OBTAIN-SUF          PCB139
EQU COMM-BLOCK          PCB1-216
EQU PGM-NAME            PCB1-8
EQU ERR-STAT            PCB9-12
EQU DBKEY               PCB13-16-B
EQU DIR-DBKEY           PCB197-200-B
EQU REC-OCCUR           PCB209-212-B
EQU DML-SEQ             PCB213-216-B
EQU OFFICE-RN           SAV1-16
EQU OFFICE-AN           SAV17-32
EQU OFFICE              WST1-90
EQU OFFICE-STREET       WST1-20
EQU OFFICE-CITY         WST21-35
EQU OFFICE-STATE        WST36-37
EQU OFFICE-ZIP          WST38-46
```

Figure 136 OPTION QUIKJOB with CA-IDMS/DB Example (Page 1 of 3)

```
EQU OFFICE-CODE         WST47-49
EQU EMPNO               WST50-55
EQU EMN                 WST56-56
EQU FNAME               WST57-68
EQU FNB                 WST69-69
EQU LNAME               WST70-84
EQU MMMMM               WST85-85
EQU SALARY              WST86-90

HDR 1A 1 DYL CALLING QUIKIDMS
HDR 1B $IPLDAT$ PAGE $PG$

*TRACE ALL
*TRACE LAST50

 8    GET   DET ATEND   10
      MOVE  DET1-80 TO PRT1
      PRINT
      GOTO 8
10    MOVE   SPACE               TO SAV1-100
      MOVE   C'QUIKIDMS'         TO PGM-NAME
      MOVE   ZEROES              TO DBKEY
      MOVE   ZEROES              TO DIR-DBKEY
      MOVE   ZEROES              TO REC-OCCUR
      MOVE   C'1'                TO DML-SEQ
      MOVE   C'OFFICE'           TO OFFICE-RN
      MOVE   C'ORG-DEMO-REGION'  TO OFFICE AN

20    CALL QJBIDMS BIND-RUN-UNIT COMM-BLOCK
             C'EMPSS01'

25    CALL QJBIDMS BIND-RN  OFFICE-RN OFFICE

30    CALL QJBIDMS READY-RETRIEVE

35    PERFORM 900 THRU 990

40    CALL QJBIDMS OBTN-1ST-RN-AN  OFFICE-RN
             OFFICE-AN  OBTAIN-SUF
      GOTO 60

50    CALL QJBIDMS OBTN-NXT-RN-AN  OFFICE-RN
             OFFICE-AN OBTAIN-SUF

60    IF ERR-STAT IS EQ TO C'0307'
      GOTO 200

70    PERFORM 900 THRU 990
90    MOVE OFFICE-STREET          TO PRT1-20
       MOVE OFFICE-CITY            TO PRT23-37
       MOVE OFFICE-STATE           TO PRT40-41
       MOVE OFFICE-ZIP             TO PRT44-52
       MOVE OFFICE-CODE            TO PRT55-57
       MOVE EMPNO                  TO PRT60-65
       MOVE EMN                    TO PRT68-68
       MOVE FNAME                  TO PRT71-82
       MOVE FNB                    TO PRT85-85
       MOVE LNAME                  TO PRT88-102
       MOVE MMMMM                  TO PRT105-105
       MOVE SALARY                 TO PRT107-112
       PRINT

100   GOTO 50

200   CALL QJBIDMS FINISH
      GOTO EOJ

900   IF ERR-STAT IS EQ TO C'0000'
      GOTO 990
      MOVE C'QUIKIDMS ABNORMAL STATUS' TO PRT1
      PRINT DOUBLESPACE
```

Figure 136 OPTION QUIKJOB with CA-IDMS/DB Example (Page 2 of 3)

```
      PRINTHEX COMM-BLOCK
      PRINTHEX OFFICE
      GOTO EOJ

990   EXIT
      END


ENTER INPUT DATA RECORDS FOR SYSDET FILE HERE.
```

Figure 136 OPTION QUIKJOB with CA-IDMS/DB Example (Page 3 of 3)

VISION:Report z/OS JCL with IMS

```
                //JOB  (your standard job statement)
                //STEP01  EXEC PGM=DFSRRC00,REGION=4500K,
                //             PARM='DLI,DYLQKIMS,PSBQREAD'
VISION:Results  //STEPLIB   DD DSN=your.RESULTS.LOADLIB,
Release 3.0     //             DISP=SHR
or later load library

VISION:Report   //           DD DSN=your.QUIKJOB.MVS.LOADLIB,
load library    //              DISP=SHR

                //           DD DSN=IMSVS.RESLIB,DISP=SHR

IMS load library //IMS        DD DSN=user.IMS.LOAD,DISP=SHR

                //IEFRDER    DD DUMMY

user's DBDLIB   //DBDLIB     DD DSN=user.IMS.LOAD,DISP=SHR
user's PSBLIB   //PSBLIB     DD DSN=user.IMS.LOAD,DISP=SHR
user's COPYLIB  //SYSCOPY    DD DSN=user.IMS.LOAD,DISP=SHR

                //SYSPRINT   DD SYSOUT=*
                //ABNLIGNR   DD DUMMY
                //SYSUDUMP   DD SYSOUT=*
                //SYS280R    DD SYSOUT=*
                //SYSOUT     DD SYSOUT=*

required        //SYS004     DD UNIT=SYSDA,SPACE=(TRK(5))
format when
running VISION:Report

required        //SORTLIB    DD DSN=SYS1.SORTLIB,DISP=SHR
only if         //SORTWK01   DD UNIT=SYSDA.SPACE=(TRK,(20,20))
SORT is used    //SORTWK02   DD UNIT=SYSDA,SPACE=(TRK,(20,20))
                //SORTWK03   DD UNIT=SYSDA,SPACE=(TRK,(20,20))

IBM IMS         //DFSRESLB   DD DSN=IMSVS.RESLIB,DISP=SHR
library

buffer          //DFSVSAMP   DD DSN=user.xx.DFSVSAMP(xxxxxx),
information     //              DISP=SHR
for             //IEFRDER    DD DUMMY
IMS             //ARMFIDX   DD DSN=user.VSAM.ARMFIDX,DISP=SHR
data base       //ARMFIMS   DD DSN=user.VSAM.ARMFIMS,DISP=SHR

for VISION:Report  //SYSDET   DD *
if applicable
                   //SYSIN    DD *
```

Figure 137 z/OS JCL with IMS Example

VISION:Report Program Calling IMS Functions

The OPTION QUIKJOB statement is a VISION:Results requirement and must be present in order for VISION:Report to run.

The OPTION SEQCHK=NO statement is the VISION:Report option, and SEQCHK=NO is only one of the operands available. If no VISION:Report option statement is entered, the default values are used (see the *Advantage VISION:Results for z/OS Reference Guide*). (WSTSIZE=5000 is required only for this sample program.)

```
OPTION QUIKJOB

OPTION SEQCHK=NO,WSTSIZE=5000

EQU STATUS-CODE          PCB11-12
EQU COUNT                WST1-4 OC ZERO

1200  CALL QJBTDLI C'GN ' C'01' WST1000
      ADD C'1' TO COUNT
      IF STATUS-CODE EQ C'GB'
      GOTO 1600
      IF COUNT GT C'0100'
      GOTO 1600
      IF STATUS-CODE IS EQ TO C'GA'  OR
      IF STATUS-CODE IS EQ TO C'GK'
      GOTO 1500
      IF STATUS-CODE IS NOT EQ TO C' '
      GOTO 1600
```

Figure 138 OPTION QUIKJOB Example

```
1500  IF COUNT GT C'0050'
      GOTO 12001550  MOVE COUNT TO PRT5-8
      MOVE STATUS-CODE TO PRT14
      MOVE WST1000-1050 TO PRT20
      PRINT
      GOTO 1200

1600  MOVE BLANK TO PRT1-80
      MOVE STATUS-CODE TO PRT30-31
      MOVE C'STATUS CODE' TO PRT15-25
      MOVE C'COUNT = ' TO PRT35-42
      MOVE COUNT TO PRT47-50
      PRINT
      READ INPUT DATA

2000  GET DET ATEND EOJ
      MOVE DET1-80 PRT1
      PRINT
      GO TO 2000
      END

ENTER DATA INPUT RECORDS FOR SYSDET FILE HERE.
```

Figure 139 OPTION QUIKJOB Example

Chapter 13: Bridge to VISION:Inquiry

If you have VISION:Inquiry r6 (or higher) installed in your data center, you can use the Bridge facility to start the VISION:Inquiry Batch query facility from VISION:Results. This enables you to use the VISION:Inquiry query syntax to easily retrieve data from IMS, DB2, and VSAM databases. It also provides the ability to automatically traverse IMS hierarchical databases and logically view them as if they were sequential.

The Bridge facility consists of new VISION:Results syntax statements, embedded VISION:Inquiry syntax statements, and the EXTRACT command described in VISION:Inquiry EXTRACT Command on page 280. The Bridge starts the VISION:Inquiry Batch Query facility to retrieve the field definitions, processes the EXTRACT command to retrieve the required fields into an extract file, and creates VISION:Results data name corresponding to the VISION:Inquiry fields. These VISION:Results data names can be subjected to the full range of data manipulation available to VISION:Results (such as LIST, arithmetic comparison, TABLE and ARRAY insertion, sampling, Letterwriter, or output to a file).

The new VISION:Results syntax statements are:

IQFILE	Defines the ddname of the extract file, contains the VISION:Inquiry syntax and EXTRACT command, and starts VISION:Inquiry to process the EXTRACT command;
IQREAD	Reads a record from the extract file.

The name of the VISION:Inquiry Batch load module can be defined by using the OPTION IQBATCH statement or the DYLINSTL parameter BATCHIQ. For OPTION statement information, see Chapter 3: Freeze and Restore Features. For information about the BATCHIQ DYLINSTL parameter, see the *Advantage VISION:Results for z/OS Installation Guide.*

VISION:Inquiry EXTRACT Command

The EXTRACT command selects and pulls data from an IMS (DL/I) database, DB2 table, or VSAM data set and writes it to a z/OS sequential data set. When EXTRACT is specified, the z/OS data set is opened, and the specified database fields are formatted into a variable length record, which is written to the opened data set.

The record format (RECFM) of the EXTRACT data set is variable blocked (VB). You must supply the block size (BLKSIZE) and logical record length (LRECL) in the data control block (DCB) for the EXTRACT data set.

The format of the record is determined by examining the inquiry. Each extracted item occupies a fixed location in the record. The fields appear left to right just as they appear in the inquiry. To facilitate their use in subsequent report generation or sorting, higher level fields in the database are repeated for each occurrence of a lower level field.

All fields are extracted in their internal form — no conversion is made. When temporary fields resulting from arithmetic operations are extracted, they are in packed format with a length of eight bytes. If the temporary field is used in an arithmetic calculation involving multiplication or division, it has four decimal places. If either addition or subtraction is involved, the temporary field has two decimal places. A temporary field assignment involving no arithmetic operations is extracted in the internal form of the assigned field. User fields defined with a user exit are not allowed and constitute a syntax error to VISION:Inquiry.

```
{EXTRACT | E} [database name] !field name!
   [LIMIT nn] [IF operand]
   ;;
```

where:

- The database name is the name of the VISION:Inquiry database. If the name is omitted, the default is the first database listed in the VISION:Inquiry directory.
- The field name can be field names, temporary fields, SUM of a field, or expression.
- LIMIT nn: nn is the maximum number of occurrences to retrieve.
- IF expression indicates the conditions under which the field is retrieved (such as EQ field2, GT field2)
- ;; are the required delimiters.

For more information about the EXTRACT command and the field names allowed, see the *Advantage VISION:Inquiry Technical Reference Guide*.

VISION:Inquiry converts the VISION:Inquiry names used in the EXRACT command according to the table shown below. The nn in the VISION:Results data names INQUIRY_TEMP_nn is a two-digit number starting with 01 for the first SUM or expression encountered in the program. It is incremented by 1 for each subsequent SUM or expression. For more information about this example, see Example on page 282.

INQUIRY TYPE	INQUIRY DATA NAME	RESULTS DATA NAME
field	PLANT.ID	PLANT_ID
temp field	%ABC	ABC
expression	PLANT.ID * 10 %DEF=SUM EMP.SAL	INQUIRY_TEMP_nn DEF
SUM	SUM EMP.SAL	INQUIRY_TEMP_nn

IQFILE Statement

The IQFILE statement defines the extract file to be created and contains the VISION:Inquiry syntax and EXTRACT command.

```
IQFILE filename [RETAIN] [VB reclen blklen] [COUNT dataname]
   [LENGTH dataname] [STATUS dataname]
   DECLARE_QUERY

   (VISION:Inquiry statements)

   END_QUERY
```

where:

- The filename is the name of the extract file for this query.
- RETAIN indicates to pass the file to a subsequent request.
- VB means the extract file is built as a VB format file. If reclen or blklen are not specified, they are taken from the JCL (ddname `filename').
- COUNT data name contains the number of records read for this file.
- LENGTH data name contains the length of the record read from the extract file.
- STATUS data name contains `E' when the last record is read from the extract file.

IQREAD Statement

The IQREAD statement reads a single record from the extract file.

```
IQREAD filename
```

JCL and Sample Program

If the VISION:Inquiry query involves an IMS database, or the VISION:Inquiry system database is an IMS database, the JCL shown in Section 12.6 should be used. If the IMS ACB (Application Control Block) feature is used, the PARM information in the illustrated JCL should be changed to:

```
//S1    EXEC    PGM=DFSRRC00,PARM=`DBB,DYL280,acbname'
```

and the ddname on the //IMS DD statement should be changed to //IMSACB DD.

If no IMS databases are involved in the query (and the VISION:Inquiry system database is a VSAM or DB2 database), the EXEC statement should be the normal VISION:Results EXEC statement.

```
//S1    EXEC    PGM=DYL280
```

Also, if no IMS databases are in use, the JCL DD statements referring to IMS RESLIB, PSBLIB, DBDLIB, or IMS databases can be removed from the JCL. If no DB2 database is involved, the JCL DD statements referring to DB2 LOADLIB can be eliminated.

Example

JCL Example:

```
//S1       EXEC PGM=DFSRRC00,PARM='DLI,DYL280,psbname',REGION=4M
//STEPLIB  DD  DSN=your.results.load.library,DISP=SHR
//         DD  DSN=your.batch.inquiry.loadlib,DISP=SHR
//         DD  DSN=IMSVS.RESLIB,DISP=SHR            (IMS RESLIB if IMS used)
//         DD  DSN=DB2TEST.DSNLOAD,DISP=SHR         (DB2 LOADLIB if DB2 used)
//IMS      DD  DSN=your.ims.psblib,DISP=SHR
//         DD  DSN=your.ims.dbdlib,DISP=SHR
//IEFRDER  DD  DUMMY,DCB=(BLKSIZE=1408,RECFM=VBS)
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYS280R  DD  SYSOUT=*
//DFSRESLB DD  DSN=IMSVS.RESLIB,DISP=SHR             (IMS RESLIB if IMS used)
//DFSVSAMP DD  *    (define VSAM buffer pools)
2048,6
4096,6
//SYS004   DD UNIT=SYSDA,SPACE=(TRK,(10,5))
//IQOUT    DD  DSN=your.inquiry.extract.file,DISP=(NEW,DELETE),
//         UNIT=DYNAM,SPACE=(TRK,(15,2),RLSE),
//         DCB=(RECFM=VB,BLKSIZE=4100,LRECL=4092)          (EXTRACT FILE)
//IQWORK   DD  DSN=your.inquiry.work.file,DISP=(NEW,DELETE),
//         UNIT=DYNAM,SPACE=(TRK,(4,2),RLSE),
//         DCB=(RECFM=FB,BLKSIZE=3200,LRECL=80)            (BRIDGE WORK FILE)
//IDIQSY20 DD  DSN=your.inquiry.system.database,DISP=SHR
//IDIQPL20 DD  DSN=INFVSAM1.DYLIQ.MCICS53.PLANT,DISP=SHR   (IMS database)
//IDIQP20O DD  DSN=INFVSAM1.DYLIQ.MCICS53.PLANTOV,DISP=SHR (IMS database)
```

Figure 140 JCL Example (Page 1 of 2)

```
//IDIQSK20 DD  DSN=INFVSAM1.DYLIQ.MCICS53.SKILL,DISP=SHR   (IMS database)
//IDIQS20O DD  DSN=INFVSAM1.DYLIQ.MCICS53.SKILLOV,DISP=SHR (IMS database)
//VSPLDS   DD  DSN=DEMO.VS.VSPLDS,DISP=SHR                (VSAM database)
//VSSKDS   DD  DSN=DEMO.VS.VSSKDS,DISP=SHR                (VSAM database)
//VSHPLDS  DD  DSN=DEMO.INQYDEMO.VSHPLDS,DISP=SHR         (VSAM database)
//VSHSKDS  DD  DSN=DEMO.INQYDEMO.VSHSKDS,DISP=SHR         (VSAM database)
//PLIDUMP  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD *
```

Figure 140 JCL Example (Page 2 of 2)

The sample program shown in this section is designed to process against an IMS database with the following structure. The sample program retrieves fields from the PLANT, EMP, and ED segments of the PLANT database.

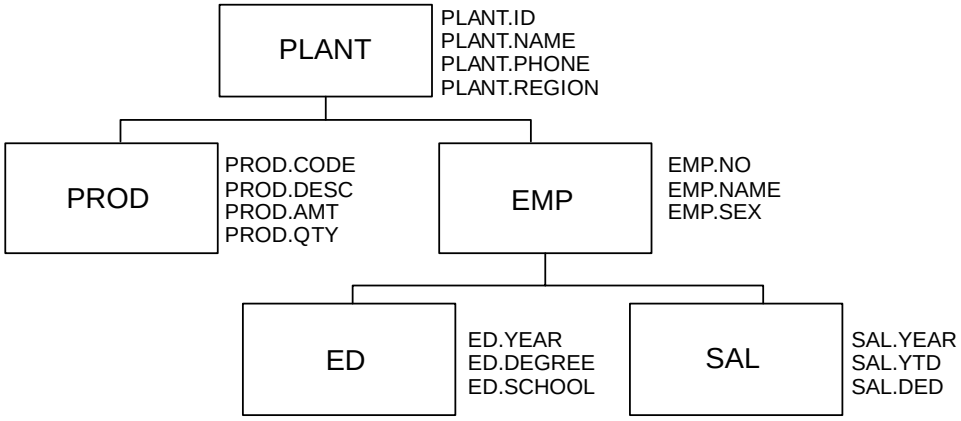

Figure 141 Sample Structure of IMS Database PLANT

Program Example:

```
1    OPTION IQBATCH IIBATCH
2    FILE SYSIN DUMMY
3    IQFILE IQOUT RETAIN  VB  4092 4100 STATUS STATFLAG  COUNT CNTR
                        LENGTH RSIZE
     *
4        DECLARE_QUERY
             /SEQ
             /IAM LTERM BATCH
             /SET TRAN IDIQ20
             EXTRACT PLANT PLANT.ID  EMP.NO  EMP.NAME
                     EMP.SEX    ED.DEGREE   ED.SCHOOL;;
5        END_QUERY
     *

       IQREAD IQOUT

       IF STATFLAG EQ 'E'
          STOP
9      ENDIF

10     LIST PLANT_ID  EMP_NO  EMP_NAME  EMP_SEX   ED_DEGREE
           ED_SCHOOL
11   ACCEPT

12   T1 'VISION:RESULTS VISION:INQUIRY BRIDGE REPORT'
```

Figure 142 Example of IMS Database PLANT (Page 1 of 2)

```
13    REPORT2
14    FILE DUMFILE DUMMY
15    USE IQOUT STATUS IQEND
16       IF IQEND EQ 'E'
17          STOP
18       ENDIF
19       PRINT PLANT_ID
20       ACCEPT
```

Figure 142 Example of IMS Database PLANT (Page 2 of 2)

The sample program processes a query against the VISION:Inquiry database PLANT. Note that the numbers to the left of the sample program are for reference purposes only. In this example, PLANT is an IMS database with the structure shown in Figure 141 on page 283. The program retrieves fields PLANT.ID, EMP.NO, EMP.NAME, EMP.SEX, ED.DEGREE, and ED.SCHOOL by automatically and transparently traversing the IMS database segments. It then lists the contents of the corresponding VISION:Results data names created: PLANT_ID, EMP_NO, EMP_NAME, EMP_SEX, ED_DEGREE, and ED_SCHOOL.

1. The optional OPTION IQBATCH statement identifies IIBATCH as the name of the VISION:Inquiry Batch load module. It can be different in your installation, and the default name could be established by the DYLINSTL parameter BATCHIQ=.
2. This statement is required only if there are no other FILE statements in the program, as in this example.
3. The IQFILE statement identifies IQOUT as the extract file. A JCL DD statement must be supplied for this file. It then starts VISION:Inquiry to retrieve the fields identified in the EXTRACT statement and writes them to the extract file. Specifying RETAIN passes the information on the extract file to a secondary VISION:Results request.
4. This defines the beginning of the VISION:Inquiry query. For complete descriptions of the query commands allowed, see the *Advantage VISION:Inquiry User Guide* and the *Advantage VISION:Inquiry Technical Reference Guide*. The VISION:Inquiry commands used in this example are:

/SEQ	Instructs VISION:Inquiry to search column 1 - 72 for syntax on the statements which follow.
/IAM LTERM	Identifies the terminal name to VISION:Inquiry. In this case, BATCH is a terminal name established in the VISION:Inquiry system database and connected to the transaction defined in the /SET TRAN statement.
/SET TRAN	Identifies the application (or transaction) IDIQ20 established in the VISION:Inquiry system database.
EXTRACT	Identifies the database to access and the fields to retrieve.

5. Defines the end of this VISION:Inquiry request.
6. Reads a record from the extract file.

7-9. Issues the STOP command when the extract file reaches the end of the file.

10. Writes a report line to SYS280R, formatting the VISION:Results data names created to represent the VISION:Inquiry fields.
13. Defines the beginning of the second VISION:Results request.

14. This statement is required only if there are no other FILE statements in the program, as in this example.

15. The USE statement causes the extract file IQOUT to be opened and read, and it sets data name IQEND as a field to test for an end-of-file condition.

16-18. Issues the STOP command when the extract file reaches the end of the file.

19. Writes the field PLANT_ID for each record read to SYSPRINT.

Report Output Example:

```
PLANT_ID    EMP_NO      EMP_NAME      EMP_SEX   ED_DEGREE  ED_SCHOOL

  10100     10103   WILLIAM AMES          M        BA      TULANE
  10100     10104   PHYLLIS LOCKMEYER     F        BA      WISCONSIN
  10100     10105   MARY ANN THOMAS       F        HS
  20150     21116   WILMA FORD            F        HS
  20150     21124   CHARLES SALTER        M        BA      EMORY
  20150     21137   PETER ZATKIN          M        BA      RUTGERS
  20150     21137   PETER ZATKIN          M        MA      TEXAS
  20150     21164   SUSAN WARE            F        BS      MICHIGAN
  30200     30201   JOHN HENRY CRANE      M        HS
  30200     30202   FREDERICH GRAY        M        BA      BOSTON U
  30200     30202   FREDERICH GRAY        M        MA      U OF MASS
```

Figure 143 Report Output Example

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z