CA JARS® Resource Accounting

Wizard Reference Guide

Release 12.7

technologies

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Technologies Product References

This document references the following CA Technologies products:

CA ACF2 for z/OS (CA ACF2)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction 13

Chapter 2: Sample Usage 19

Chapter 1: Introduction

This guide contains all of the information needed to use CA JARS Wizard. However, you are expected to have some familiarity with your CA product including the files it creates and its operational characteristics.

Note: Throughout this guide, examples are provided to illustrate various control statements. Many of these examples contain sample data.Your data may differ from the data that is displayed in this guide.

CA JARS Wizard is a generalized, computational report generator used to process data produced by other CA products. User-designed reports can be created to display detail, summary, and exception system information in minutes.

File handling characteristics include:

- Predefined file processing

  Predefined (glossary-defined) files can be processed.

- Controlled file modification

  The contents of input files cannot be modified; all update functions require a user-specified output file. Output from product processing can be written to a new data file.

- File merging

  Data from one file can be merged with another.

While processing data, this product can:

- generally complete all processing in one pass of the program
- perform automatic arithmetic functions and user-defined calculations
- delete, transform, or modify entire records or specified data within records
- select or reject certain records for processing
- sort and total data on reports for user-defined fields

CA JARS Wizard allows great flexibility in report design. Based on the available report types, you can create your own reports by indicating what data you want used and in what format it is to be displayed.The following report types can be produced:

- Detail or Summary Reports

  Detail Reports

  print one line of data from each record. They can include group totals after each set of related records.

  Summary Reports

  suppress the printing of detail lines and print only the group total lines.

- Exception Reports

  Exception reports can be of any length, but are designed to be short representations of records that meet unusual conditions. For example, you may want to print all job steps that have a cancellation code indicating operator intervention.This product can produce any number of these reports in a single run.

- Ranked Record Reports

  Ranked record reports show summary records ranked by user-specified criteria. For example, an installation may consolidate all of the job step records for a week into a single summary record for a client user department. These summary records can then be listed by user-specified criteria, such as total run time, printer forms usage, etc. CA JARS Wizard can produce up to ten ranked reports in a single run.

- Calendar Reports

  Calendar reports display summary data in calendar format. One page is generated for each month. A maximum of eight summary data fields can be displayed for each day.

- Plot Reports

  Plot reports are graphic representations of numeric values.

- Cross-tabulation Reports

  Cross-tabulation reports print tables of user-specified data.

Detailed descriptions of this product and instructions for its use are described in the remaining chapters of this guide.

This section contains the following topics:

Operations

The rest of this chapter describes the z/OS JCL necessary to:

- describe input to this product
- execute this product
- create a listing of the definitions in a glossary

Input used by CA JARS Wizard

There are two categories of input accepted by this product for reporting. They are:

- CA JARS created history or summary files
- User created or defined data

CA JARS created data includes glossary formatted:

- History level 6 or level 7 detail files
- summary level 1 through 5 files
- CA JARS Interface output

  Note: For more information,see the "Using CA JARS with Special Environments" chapter in the *CA JARS Systems Programmer Guide.*

The user created data can be secondary input used as lookup tables or primary input files. Refer to the appropriate chapters in this guide for information on functional characteristics and input statements. User data must be in flat sequential data sets.

The following input is not accepted:

- CA JARS account files
- VSAM keyed (KSDS) or Relative Record (RRDS) data sets
- Indexed files
- Raw SMF files

Executing CA JARS Wizard

Use the following sample JCL to create CA JARS Wizard reports and output:

```
//WIZARD   JOB ,,CLASS=X,MSGCLASS=X
//WIZSTEP  EXEC PGM=WIZARD,REGION=2048K
//STEPLIB  DD   DSN=CAI.CAJRLOAD,DISP=SHR
//CAIJWCP  DD   DSN=CAI.CAJROPTN,DISP=SHR
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD   UNIT=SYSDA,SPACE=(TRK,(30,5))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(TRK,(30,5))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(TRK,(30,5))
//SORTWK04 DD   UNIT=SYSDA,SPACE=(TRK,(30,5))
//CAIJWI   DD   DSN=CAI.JARS.WIZARD.DATA,DISP=SHR
//CAIJWOP  DD   UNIT=SYSDA,SPACE=(CYL,(10,2))
//SYSPRINT DD   SYSOUT=*,DCB=(BLKSIZE=133,LRECL=133,RECFM=FBA)
//SYSOUT   DD   SYSOUT=*
//SORTMSG  DD   SYSOUT=*
//SYSIN    DD   *
Insert CA JARS Wizard control statements here -
/*
//
```

The following table lists the ddnames and their functions in a CA JARS Wizard run.

CA JARS Wizard Function	DDname
Primary input file	CAIJWI
Secondary input file #1	CAIJWI2
Secondary input file #2	CAIJWI3
Secondary input file #3	CAIJWI4
Secondary file for concatenation (UPD)	CAIJWOL
Output file and the interim file for RANK, EXCEPTION, and LIST(BOTH)	CAIJWOP CAIJWOP
SORT library	SORTLIB
SORT message data set	SORTMSG
SORT work files (minimum of 3)	SORTWK01 SORTWK02 SORTWK03 &bul. &bul.
CA JARS Wizard Source Statement Library	CAIJWCP

CA JARS Wizard Function	DDname
Report output	SYSPRINT DCB=(RECFM=FBA)
Control statements input	SYSIN
System message data set	SYSOUT

Some CA JARS Wizard functions require that additional ddnames be included in the JCL for the run. Here is a list of these functions and the ddnames they require:

CA JARS Wizard Function	DDname Required
Output Update: OUTPUT(UPD)	CAIJWOL
File Matching	CAIJWI2-4 (as necessary)
COPY Statement	CAIJWCP
Output File Creation	CAIJWOP
Ranking Reports (RANK Statement)	CAIJWOP
Exception Reports (EXCEPTION Statement)	CAIJWOP
LIST(BOTH) Reports	CAIJWOP

Creating CA JARS Wizard Glossary Listings

Many glossaries exist for CA JARS Wizard. Each describes the locations and attributes of various fields used throughout the various product systems. See Appendix D, E, F and G for a complete list of the glossaries provided with your product.

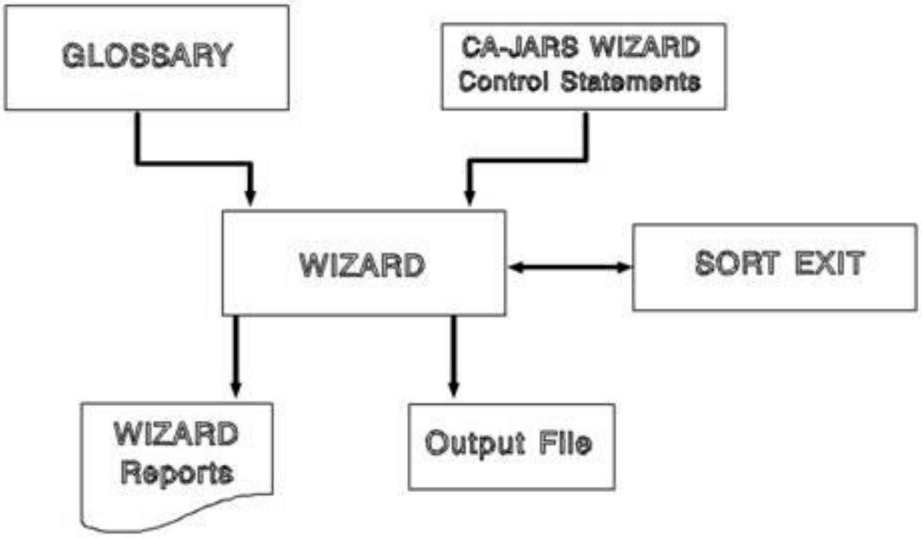
The definitions included in any available CA JARS Wizard glossary can be listed using this JCL:

```
//GLOSSARY JOB MSGLEVEL=1
//STEP1 EXEC PGM=WIZPG
//STEPLIB DD DSN=CAI.CAJRLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
glossary name
/*
//
```

You can specify up to 32 glossary names in any run, but each glossary name must be on a new record.

The report generated by the WIZPG program describes the positions, length, format, and default suffixes for every field name given in the glossary. Overlapping fields and unused positions are also identified.

CA JARS Wizard Processing Flowchart



Chapter 2: Sample Usage

This chapter contains CA JARS Wizard sample reports created using the CA JARS glossary CAWGJR70. It begins with a simple report and progresses through a variety of different report types and CA JARS Wizard features.

This chapter is an exceptionally helpful aid for *new* CA JARS Wizard users. No attempt has been made to demonstrate either all available features or all functional variations. In addition, while CA JARS Wizard automatically prints standard heading lines, we have selectively omitted these in some reports to capture the main portion of the report and fit it appropriately on the page.

Note: For more information,see the "Control Statements (see page 59)" chapter for detailed information on the control statements, and CA JARS Wizard facilities used.

This section contains the following topics:

Creating Sample Reports

You can create each of the reports in this section using the sample JCL described in the "Introduction (see page 13)" chapter. The control statements for each report in this chapter are stored in the CAJROPTN library and are distributed with CA JARS. In addition, special execution considerations for sample reports are described, wherever applicable, throughout this chapter.

The table on the following page lists each CA JARS Wizard reporting feature discussed in this chapter, the title(s) of the CA JARS Wizard report(s) produced, and the name of the member containing the control statements for the run.

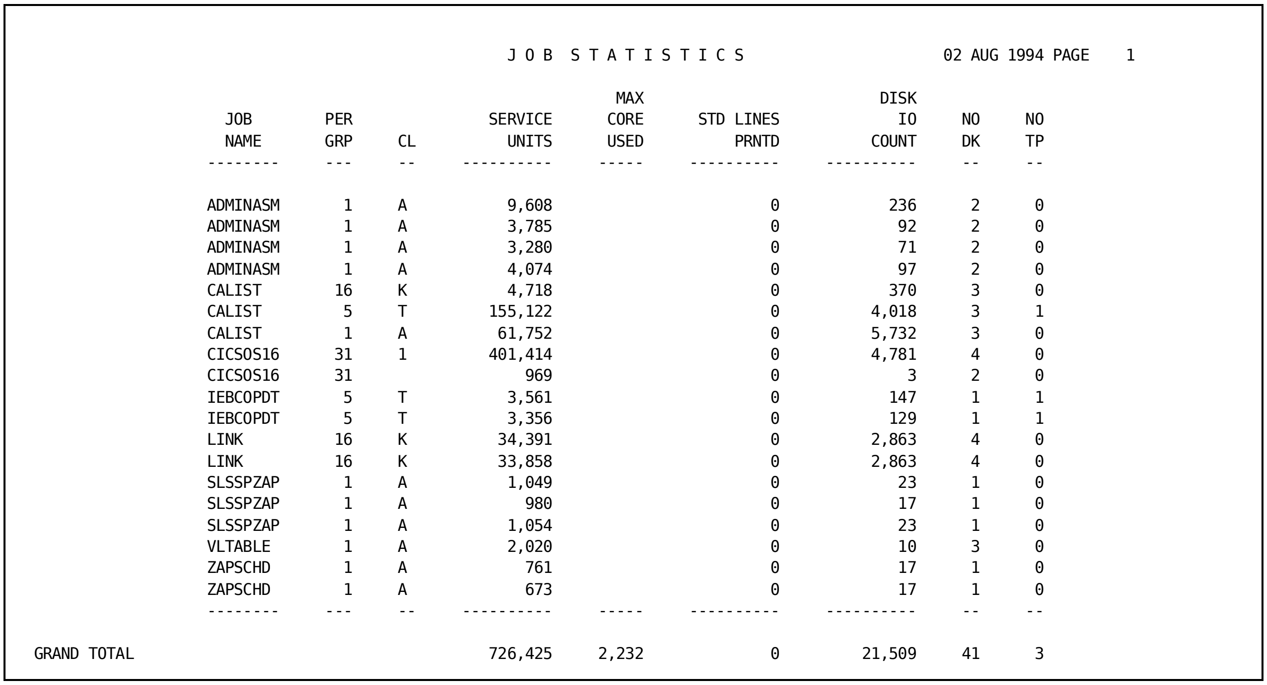
CA JARS Wizard Feature	Control Statement Filename	Report Title
Basic Report Listing	JARSWZ01	JOB STATISTICS
Selecting Records	JARSWZ02	JOB STATISTICS FOR JOBS BEGINNING WITH 'ADMIN'
Control Breaks and Calculations	JARSWZ03	JOB STATISTICS WITH JOB TOTALS
Summary Report	JARSWZ04	JOB STATISTICS SUMMARIZED
Page, Line, and Column Headings	JARSWZ05	CA JARS REPORTING JOB STATISTICS REPORT
Ranking Reports	JARSWZ06	RANKING REPORT A - SORTED BY ELAPSED TIME RANKING REPORT B - SORTED BY CPU TIME
Exception Reports	JARSWZ07	REPORT CORE ALLOC OVER 40000 REPORT CORE USED OVER 3000
Calendar Report	JARSWZ08	SEPTEMBER 1985
Plot Report	JARSWZ09	I/O ACTIVITY ANALYSIS
File Matching and Table Lookups	JARSWZ10	FILE MATCHING: COST CENTER ANALYSIS
Decode Function	JARSWZ11	COST CENTER ANALYSIS
Input File Update	JARSWZ12	execution statistics only
Extract File Creation	JARSWZ13	execution statistics only
Summary File Creation	JARSWZ14	execution statistics only
Output Merge Function	JARSWZ15	execution statistics only
New File Creation	JARSWZ16	execution statistics only

JARSWZ01: Basic Report Listing

JARSWZ01 produces a report showing job statistics by job name. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are listed. Data is given alphabetically by job name.

The only processing function executed in this example is to sort the records by job name.The sort can be eliminated if the file is already in the desired sequence.

The CA JARS input files tend to be very large, and selection of records for processing must be considered when designing a CA JARS Wizard program. Some of the remaining examples in this chapter show how to select and summarize records to obtain data in a more useful and less lengthy format.



```
                                           J O B  S T A T I S T I C S                 02 AUG 1994 PAGE    1

                                                             MAX                      DISK
                   JOB        PER                SERVICE     CORE    STD LINES          IO     NO     NO
                   NAME       GRP     CL           UNITS     USED        PRNTD       COUNT     DK     TP
                 --------     ---     --      ----------    -----   ----------  ----------     --     --

                 ADMINASM       1     A            9,608                     0         236      2      0
                 ADMINASM       1     A            3,785                     0          92      2      0
                 ADMINASM       1     A            3,280                     0          71      2      0
                 ADMINASM       1     A            4,074                     0          97      2      0
                 CALIST        16     K            4,718                     0         370      3      0
                 CALIST         5     T          155,122                     0       4,018      3      1
                 CALIST         1     A           61,752                     0       5,732      3      0
                 CICSOS16      31     1          401,414                     0       4,781      4      0
                 CICSOS16      31                    969                     0           3      2      0
                 IEBCOPDT       5     T            3,561                     0         147      1      1
                 IEBCOPDT       5     T            3,356                     0         129      1      1
                 LINK          16     K           34,391                     0       2,863      4      0
                 LINK          16     K           33,858                     0       2,863      4      0
                 SLSSPZAP       1     A            1,049                     0          23      1      0
                 SLSSPZAP       1     A              980                     0          17      1      0
                 SLSSPZAP       1     A            1,054                     0          23      1      0
                 VLTABLE        1     A            2,020                     0          10      3      0
                 ZAPSCHD        1     A              761                     0          17      1      0
                 ZAPSCHD        1     A              673                     0          17      1      0
                 --------     ---     --      ----------    -----   ----------  ----------     --     --

 GRAND TOTAL                                     726,425    2,232            0      21,509     41      3
```

JARSWZ01 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
LIST JOB/NAME, PER/GRP, CL, SERVICE/UNITS, CORE/USED(MX),
     STD LINES/PRNTD, DISK/IO/COUNT, NO/DK, NO/TP
HEADING1C J O B  S T A T I S T I C S
```

JARSWZ01 Execution Description

The CA JARS Wizard control statements in this example function as follows:

INPUT

This INPUT statement identifies the CA JARS Wizard glossary for the primary input file for the run. All the files to be processed by CA JARS Wizard must be defined. Refer to Chapter 4 for information concerning record type qualification. The glossary definition describes each field in the input record with accessing, processing, and printing attributes and options. A field in a file, predefined in the glossary, can be accessed, processed, and printed by reference to the glossary field name.

SELECT

The SELECT statement instructs CA JARS Wizard to examine the PRO ID field and the STEP JOB IND field. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are processed. For further information on record types and data contents see "The CAWGJR70 Glossary" Appendix chapter. This statement is required by CA JARS Wizard for certain input files, since such files contain are large or contain records in many different formats. Qualifying record selection ensures that the proper records and/or data fields are accessed.

SORT

The SORT statement identifies the sorting sequence prior to processing. If the SORT statement is omitted, records are processed in the same order that they are read from the file.

In this example records are sorted only by JOB NAME. Consequently, records are listed alphabetically by job name.

LIST

LIST identifies the data fields to be printed and the order, (defined from left to right), they are to be printed in. CA JARS Wizard performs automatic report formatting. In this example, CA JARS Wizard prints automatic standard headings. These can be overridden through your specifications.

Each field name in the LIST statement is printed and CA JARS Wizard automatically edits each data field. The automatic editing functions include printing a column heading, converting the data from internal format to display format, and inserting periods (.) in the time to separate the seconds and milliseconds. Your product glossary contains basic information that specifies to CA JARS Wizard the automatic processing and editing functions that should be performed. For example, CA JARS Wizard uses the field name defined in the glossary as the column heading. If you include a slash in the field name, the column heading is split over two lines. For example, the field name SERVICE/UNITS prints as:

```
SERVICE
UNITS
```

CA JARS Wizard automatically computes and prints grand totals for numeric fields in the LIST statement. Totals are not printed for numeric fields when the total value would be meaningless. For example, grand totals are not calculated for the field JOB NAME, (the name of a job).

HEADING

The HEADING statement specifies headings for the report. The statement given in this example indicates that the heading should be centered on line 1 (1C) and should read: J O B S T A T I S T I C S.

Other statements and/or methods that can be used to override the field editing and automatic column heading functions are described in the "Functional Characteristics (see page 119)" chapter.The following examples show how to override or supplement these automatic functions.

JARSWZ02: Selecting Records

JARSWZ02 produces a report that lists statistics for jobs with job names beginning with ADMIN.

This example shows how to select records in an input file. It produces the same basic listing as in JARSWZ01. However, data is restricted to records that meet the three special conditions listed below, bypassing all other records in the input file.

1. The processing ID of the record is S (SMF) or T (TSO)

2. STEP JOB IND is 1 (a job record) or 11 (a one-step job record)
3. The first five characters of the job name are ADMIN

```
                          JOB STATISTICS FOR JOBS BEGINNING WITH 'ADMIN'          02 AUG 1994 PAGE    1

                                                           MAX                         DISK
                    JOB         PER             SERVICE   CORE      STD LINES           IO      NO     NO
                    NAME        GRP     CL        UNITS   USED          PRNTD        COUNT      DK     TP
                  --------      ---     --   ----------   -----    ----------   ----------      --     --

                  ADMINASM        1     A         9,608                     0          236       2      0
                  ADMINASM        1     A         3,785                     0           92       2      0
                  ADMINASM        1     A         3,280                     0           71       2      0
                  ADMINASM        1     A         4,074                     0           97       2      0
                  --------      ---     --   ----------   -----    ----------   ----------      --     --

GRAND TOTAL                                      20,747   1,024             0          496       8      0
```

JARSWZ02 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
SELECT(S) JOB/NAME(1,5) EQ 'ADMIN'
LIST JOB/NAME, PER/GRP, CL, SERVICE/UNITS, CORE/USED(MX),
     STD LINES/PRNTD, DISK/IO/COUNT, NO/DK, NO/TP
HEADING1C JOB STATISTICS FOR JOBS BEGINNING WITH 'ADMIN'
```

JARSWZ02 Execution Description

The control statements in this example are similar to and function in the same way as described in JARSWZ01. A SELECT statement has been added to choose only jobs with job names beginning with ADMIN.

SELECT

Two SELECT statements are shown for this run. The first SELECT statement is also used in report JARSWZ01 and is described there. The second SELECT statement instructs CA JARS Wizard to examine the JOB NAME field in each record and select a record for processing only if the job name begins with ADMIN. The (S) suffix code appended to the second SELECT indicates that the selection is to be performed after the SORT.

Only the selected records are sorted and processed. Again, CA JARS Wizard provides all report formatting, data conversions, and grand totals automatically.

Sometimes it is easier to be negative. The second selection condition in this example could also be specified as follows:

```
REJECT(S) JOB/NAME(1,5) NE 'ADMIN'
```

The report produced using this control statement and the first SELECT statement would be identical to the report produced by JARSWZ02.

JARSWZ03: Control Breaks and Calculations

JARSWZ03 produces a report showing job statistics by job name. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are listed. Data is given alphabetically by job name.

All processed records are organized into control groups by job name. Automatic subtotals for numeric data fields in each control group are calculated. The use of a COMPUTE statement in a CA JARS Wizard run is also demonstrated.

```
                                      J O B  S T A T I S T I C S  W I T H  J O B  T O T A L S           08 AUG 1994 PAGE    1

                                             MAX                          DISK                   ELAPSED       ACTIVE          CPU
  JOB        PER               SERVICE      CORE      STD LINES             IO     NO     NO        TIME         TIME         TIME
  NAME       GRP     CL          UNITS      USED          PRNTD          COUNT     DK     TP    HH.MM.SS     HH.MM.SS     HH.MM.SS
--------     ---     --     ----------     -----     ----------     ----------     --     --    --------     --------     --------

ADMINASM       1     A           4,074                        0             97      2      0    00.00.21     00.00.16     00.00.01
ADMINASM       1     A           3,280                        0             71      2      0    00.00.16     00.00.10     00.00.01
ADMINASM       1     A           3,785                        0             92      2      0    00.00.17     00.00.12     00.00.01
ADMINASM       1     A           9,608                        0            236      2      0    00.00.51     00.00.45     00.00.03
                            ----------     -----     ----------     ----------     --     --    --------     --------     --------
ADMINASM                        20,747     1,024              0            496      8      0    00.01.45     00.01.23     00.00.06
JOB NAME TOTAL

CALIST        16     K           4,718                        0            370      3      0    00.00.12     00.00.09     00.00.01
CALIST         5     T         155,122                        0          4,018      3      1    00.13.14     00.10.38     00.00.52
CALIST         1     A          61,752                        0          5,732      3      0    00.03.36     00.03.33     00.00.14
                            ----------     -----     ----------     ----------     --     --   --------     --------     --------
CALIST                         221,592       360              0         10,120      9      1    00.17.02     00.14.20     00.01.07
JOB NAME TOTAL

CICSOS16      31     1         401,414                        0          4,781      4      0    44.29.54     44.29.49     00.02.15
CICSOS16      31                   969                        0              3      2      0    00.00.02     00.00.01
                            ----------     -----     ----------     ----------     --     --    --------     --------     --------
CICSOS16                       402,383     2,232              0          4,784      6      0    44.29.56     44.29.50     00.02.15
JOB NAME TOTAL

IEBCOPDT       5     T           3,561                        0            147      1      1    00.02.38     00.02.37     00.00.01
IEBCOPDT       5     T           3,356                        0            129      1      1    00.06.06     00.06.05     00.00.01
                            ----------     -----     ----------     ----------     --     --    --------     --------     --------
IEBCOPDT                         6,917       768              0            276      2      2    00.08.44     00.08.42     00.00.02
JOB NAME TOTAL

LINK          16     K          34,391                        0          2,863      4      0    00.01.24     00.01.21     00.00.08
LINK          16     K          33,858                        0          2,863      4      0    00.01.33     00.01.30     00.00.08
                            ----------     -----     ----------     ----------     --     --    --------     --------     --------
LINK                            68,249       264              0          5,726      8      0    00.02.57     00.02.51     00.00.16
JOB NAME TOTAL

SLSSPZAP       1     A             980                        0             17      1      0    00.00.03     00.00.02
SLSSPZAP       1     A           1,049                        0             23      1      0    00.00.03     00.00.02
SLSSPZAP       1     A           1,054                        0             23      1      0    00.00.04     00.00.03
                            ----------     -----     ----------     ----------     --     --    --------     --------     --------
SLSSPZAP                         3,083        48              0             63      3      0    00.00.10     00.00.07
JOB NAME TOTAL

VLTABLE        1     A           2,020                        0             10      3      0    00.00.04     00.00.03     00.00.01
                            ----------     -----     ----------     ----------     --     --    --------     --------     --------
VLTABLE                          2,020     1,024              0             10      3      0    00.00.04     00.00.03     00.00.01
JOB NAME TOTAL

ZAPSCHD        1     A             673                        0             17      1      0    00.00.02     00.00.01
ZAPSCHD        1     A             761                        0             17      1      0    00.00.01     00.00.01
                            ----------     -----     ----------     ----------     --     --    --------     --------     --------
ZAPSCHD                          1,434        36              0             34      2      0    00.00.03     00.00.02
JOB NAME TOTAL

  GRAND TOTAL                  726,425     2,232              0         21,509     41      3    45.00.41     44.57.18     00.03.47
  GRAND TOTAL                  726,425     2,232              0         21,509     41      3    45.00.41     44.57.18     00.03.47
```

JARSWZ03 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
BREAK JOB/NAME
COMPUTE ELAPSED/TIME EQ (ELAPSED/TIME/HRS * 3600)
COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/HRS * 3600)
COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600)
LIST JOB/NAME, PER/GRP, CL, SERVICE/UNITS, CORE/USED(MX),
     STD LINES/PRNTD, DISK/IO/COUNT, NO/DK, NO/TP, ELAPSED/TIME(HS),
     ACTIVE/TIME(HS), CPU/TIME(HS)
HEADING1C J O B  S T A T I S T I C S  W I T H  J O B  T O T A L S
```

JARSWZ03 Execution Description

The INPUT, SELECT, and SORT statements function as described previously. Other control statements function as follows:

BREAK

The BREAK statement defines logical groups of records. The group defined in this example is comprised of records with the same job name. A control break is performed with each change in the JOB/NAME field. Control break totals are provided for the records in each control group. The control break that occurs on changes in JOB/NAME is referred to as the lowest, or first level control break. In this example the JOB/NAME break is also the highest level control break because it is the only control break.

When using the BREAK statement in a CA JARS Wizard run, you:

- **must** include a SORT statement in the run. Fields that are not listed in the SORT statement **may not** be specified in the BREAK statement.
- **must** list the BREAK statement fields in the same order as they are listed in the SORT statement.
- **must not** list more fields in the BREAK statement than are given in the SORT statement. (The BREAK statement may, however, include less fields than the SORT statement).

COMPUTE

In this example, the COMPUTE statements convert ELAPSED/TIME/HRS, ACTIVE/TIME/HRS, and CPU/TIME/HRS from hours to seconds. This is done as a preliminary step to displaying each of these fields in HH.MM.SS format on the report. To display a time field in HH.MM.SS format, the (HS) suffix code is appended to the field in the LIST statement. However, the (HS) suffix requires that the time field data be stored in seconds.

LIST

LIST defines the fields to be included in the printed output line.

In conjunction with the BREAK statement, the LIST statement causes CA JARS Wizard to perform automatic computations. It adds all numeric fields specified in the LIST statement and maintains an accumulator for each control level group. In this example, CA JARS Wizard maintains a job name accumulator and a grand total accumulator for each of the numeric fields defined in the LIST statement. The contents of these accumulators are printed at the designated control break.

Some reports may contain fields that do not have meaningful accumulated totals. This is true in our example for PER/GRP and CL. For these fields control break totals can be suppressed using the (NT) suffix. This suffix can be specified in the glossary for the particular field definition, or as a suffix to the field name in the LIST statement.

CA JARS Wizard provides automatic control break headings. The general form of control break headings is: field name TOTAL. In this example, the control break heading is JOB NAME TOTAL. You can override these control break headings. See the discussion of the BREAK statement in the "Control Statements (see page 59)" chapter.

JARSWZ04: Summary Report

JARSWZ04 creates a report that summarizes the report produced in JARSWZ03. Only the control break and grand totals are printed; detail print lines are suppressed. Records are selected and processed as usual, but the print step is bypassed until it reaches a control break.

```
                                            J O B  S T A T I S T I C S  S U M M A R I Z E D              08 AUG 1994 PAGE    1

                                             MAX                     DISK                        ELAPSED     ACTIVE          CPU
  JOB         PER               SERVICE     CORE     STD LINES         IO      NO      NO          TIME       TIME          TIME
  NAME        GRP      CL         UNITS     USED         PRNTD      COUNT      DK      TP      HH.MM.SS   HH.MM.SS      HH.MM.SS
--------      ---      --    ----------    -----    ----------   ----------    --      --      --------   --------      --------

ADMINASM                         20,747    1,024             0        496       8       0      00.01.45   00.01.23      00.00.06
CALIST                          221,592      360             0     10,120       9       1      00.17.02   00.14.20      00.01.07
CICSOS16                        402,383    2,232             0      4,784       6       0      44.29.56   44.29.50      00.02.15
IEBCOPDT                          6,917      768             0        276       2       2      00.08.44   00.08.42      00.00.02
LINK                             68,249      264             0      5,726       8       0      00.02.57   00.02.51      00.00.16
SLSSPZAP                          3,083       48             0         63       3       0      00.00.10   00.00.07
VLTABLE                           2,020    1,024             0         10       3       0      00.00.04   00.00.03      00.00.01
ZAPSCHD                           1,434       36             0         34       2       0      00.00.03   00.00.02
--------      ---      --    ----------    -----    ----------   ----------    --      --      --------   --------      --------

  GRAND TOTAL                   726,425    2,232             0     21,509      41       3      45.00.41   44.57.18      00.03.47
```

JARSWZ04 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
BREAK JOB/NAME
COMPUTE ELAPSED/TIME EQ (ELAPSED/TIME/HRS * 3600)
COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/HRS * 3600)
COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600)
LIST(SUM) JOB/NAME, PER/GRP, CL, SERVICE/UNITS, CORE/USED(MX),
     STD LINES/PRNTD, DISK/IO/COUNT, NO/DK, NO/TP, ELAPSED/TIME(HS),
     ACTIVE/TIME(HS), CPU/TIME(HS)
HEADING1C J O B  S T A T I S T I C S  S U M M A R I Z E D
```

JARSWZ04 Execution Description

The control statements in this example function as described in earlier examples. This example, however constructs a single summary record for all detail records having the same job name.

LIST(SUM)

The (SUM) suffix has been added to the LIST statement to request a summary report instead of a detail listing.

Summary records listed in a summary report may include consolidated data from any number of detail records.

JARSWZ05: Page, Line, and Column Headings

JARSWZ05 adds extensive page, line, and column headings to the report produced in JARSWZ03. It displays job statistics by job name as does JARSWZ03, but HEADING and TITLE statements usage is demonstrated. Compare the report produced in JARSWZ03 with the report produced in this example to identify all of the differences between the reports.

```
CA                                               C A - J A R S   R E P O R T I N G                           JARSWZ05 PAGE      1
WIZARD REPORT WRITER                                                                                  08 AUG 1994 11.01.44

                                              J O B   S T A T I S T I C S   R E P O R T

                                                                               *-NUMBER OF--*
                                            MAX                        DISK                      ELAPSED      ACTIVE         CPU
     JOB       PER             SERVICE     CORE     STD LINES            IO                         TIME        TIME        TIME
     NAME      GRP    CL         UNITS     USED         PRNTD         COUNT    DISKS    TAPES  HH.MM.SS    HH.MM.SS    HH.MM.SS
   --------    ---    --    ----------    -----    ----------    ----------    -----    -----  --------    --------    --------

   ADMINASM      1    A          4,074                      0            97        2        0  00.00.21    00.00.16    00.00.01
   ADMINASM      1    A          3,280                      0            71        2        0  00.00.16    00.00.10    00.00.01
   ADMINASM      1    A          3,785                      0            92        2        0  00.00.17    00.00.12    00.00.01
   ADMINASM      1    A          9,608                      0           236        2        0  00.00.51    00.00.45    00.00.03
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   ADMINASM                     20,747    1,024             0           496        8        0  00.01.45    00.01.23    00.00.06
JOB NAME TOTAL

   CALIST       16    K          4,718                      0           370        3        0  00.00.12    00.00.09    00.00.01
   CALIST        5    T        155,122                      0         4,018        3        1  00.13.14    00.10.38    00.00.52
   CALIST        1    A         61,752                      0         5,732        3        0  00.03.36    00.03.33    00.00.14
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   CALIST                      221,592      360            0         10,120        9        1  00.17.02    00.14.20    00.01.07
JOB NAME TOTAL

   CICSOS16     31    1        401,414                      0         4,781        4        0  44.29.54    44.29.49    00.02.15
   CICSOS16     31                 969                      0             3        2        0  00.00.02    00.00.01
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   CICSOS16                    402,383    2,232             0         4,784        6        0  44.29.56    44.29.50    00.02.15
JOB NAME TOTAL

   IEBCOPDT      5    T          3,561                      0           147        1        1  00.02.38    00.02.37    00.00.01
   IEBCOPDT      5    T          3,356                      0           129        1        1  00.06.06    00.06.05    00.00.01
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   IEBCOPDT                      6,917      768             0           276        2        2  00.08.44    00.08.42    00.00.02
JOB NAME TOTAL

   LINK         16    K         34,391                      0         2,863        4        0  00.01.24    00.01.21    00.00.08
   LINK         16    K         33,858                      0         2,863        4        0  00.01.33    00.01.30    00.00.08
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   LINK                         68,249      264             0         5,726        8        0  00.02.57    00.02.51    00.00.16
JOB NAME TOTAL

   SLSSPZAP      1    A            980                      0            17        1        0  00.00.03    00.00.02
   SLSSPZAP      1    A          1,049                      0            23        1        0  00.00.03    00.00.02
   SLSSPZAP      1    A          1,054                      0            23        1        0  00.00.04    00.00.03
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   SLSSPZAP                      3,083       48             0            63        3        0  00.00.10    00.00.07
JOB NAME TOTAL

   VLTABLE       1    A          2,020                      0            10        3        0  00.00.04    00.00.03    00.00.01
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   VLTABLE                       2,020    1,024             0            10        3        0  00.00.04    00.00.03    00.00.01
JOB NAME TOTAL

   ZAPSCHD       1    A            673                      0            17        1        0  00.00.02    00.00.01
   ZAPSCHD       1    A            761                      0            17        1        0  00.00.01    00.00.01
                            ----------    -----    ----------    ----------    -----    -----  --------    --------    --------
   ZAPSCHD                       1,434       36             0            34        2        0  00.00.03    00.00.02
JOB NAME TOTAL

  GRAND TOTAL                  726,425    2,232             0        21,509       41        3  45.00.41    44.57.18    00.03.47
```

JARSWZ05 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
BREAK JOB/NAME
COMPUTE ELAPSED/TIME EQ (ELAPSED/TIME/HRS * 3600)
COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/HRS * 3600)
COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600)
LIST JOB/NAME, PER/GRP, CL, SERVICE/UNITS, CORE/USED(MX),
     STD LINES/PRNTD, DISK/IO/COUNT, NO/DK'DISKS', NO/TP'TAPES',
     ELAPSED/TIME(HS), ACTIVE/TIME(HS), CPU/TIME(HS)
HEADING1L CA
HEADING1C C A - J A R S   R E P O R T I N G
HEADING1R JARSWZ05 PAGE=WIZPAGES
HEADING2L WIZARD REPORT WRITER
HEADING2R =TODAY =TOD(HS)
HEADING4C J O B  S T A T I S T I C S  R E P O R T
TITLE 8-9 = NUMBER OF
```

JARSWZ05 Execution Description

The INPUT, SELECT, SORT, BREAK, and COMPUTE statements are executed as discussed previously. The other statements function as follows:

LIST

The LIST statement in this example provides column titles for the NO/DK and NO/TP fields of the report. These new column titles work in conjunction with the TITLE statement heading to produce more meaningful column headings for these two fields.

HEADING nx

Three lines of heading text are supplied (for lines 1, 2, and 4). They are printed on the left, center, or right side of the page. For example, HEADING1L applies to the first line on the left side.

Note: No HEADING statement is provided for the third, fifth, or sixth lines of the heading text. Line 3 prints as blank, lines 5 and six do not print at all.

TITLE

Text to be printed above columns 8 and 9 (the NO/DK and NO/TP columns) is specified. CA JARS Wizard supplies dashes to ensure that the title is centered and has the proper width, when necessary.

Keywords

TODAY, TOD, and WIZPAGES are keywords. TODAY and TOD contain the current date and the time of day that the report is generated. WIZPAGES contains the current page number of the report.

Where no user headings are indicated, standard headings are always printed. For user-specified headings, the current date and page number appear on the right-hand side of the first line of each page, unless overridden by a HEADING1R statement.

JARSWZ06: Ranking Reports

The top 24 transactions of CPU time and elapsed time are displayed in JARSWZ06. The first report lists the top 24 users of elapsed time; the second report lists the top 24 users of CPU time. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are listed. Data is presented alphabetically by job name.

The JARSWZ06 reports are produced using the ranking facility. Ranking is a two step process. In the first step, CA JARS Wizard performs normal processing to create summary records as it would for a summary report. In the second step, CA JARS Wizard sorts the summary records as required for one or more ranking reports.

```
                                 RANKING REPORT A - SORTED BY ELAPSED TIME                         07 AUG 1994 PAGE    1

                      ELAPSED       ACTIVE          CPU
          JOB            TIME         TIME         TIME                    TAPE                    DISK
          NAME       HH.MM.SS     HH.MM.SS     HH.MM.SS                IO COUNT                IO COUNT
        --------     --------     --------     --------     -------------------     -------------------

        CHOMS       149.55.50     38.15.43     01.06.52                       0                   1,248
        CICSOS16     88.59.52     88.59.41     00.04.30                       0                   9,568
        ACSSPDBC     66.19.00     66.18.58     00.11.51                       0                  17,188
        OPERMON      37.03.27     00.23.45     00.01.31                       0                   6,794
        PCSSPDBC     17.45.32     17.45.31     00.01.59                   3,055                   2,975
        BRUCE        09.24.13     00.20.15     00.05.53                       0                  12,753
        TSOJFL       09.20.03     00.27.34     00.02.27                       0                   5,338
        DON          08.20.57     00.35.01     00.06.37                       0                  46,874
        ANDYB        07.39.25     00.32.01     00.02.59                       0                  20,022
        DANK         07.37.52     00.51.38     00.11.09                       0                   8,214
        FRANKIE      07.02.18     00.44.10     00.08.21                       0                  72,889
        TSOSRM       06.48.36     00.02.05     00.00.37                       0                     686
        GARY         06.37.35     00.15.05     00.00.59                       0                   1,211
        DONU         06.36.17     00.19.13     00.03.23                       0                   1,602
        QARICH       06.36.13     00.02.17     00.00.34                       0                     768
        CHUCK        06.33.55     00.11.43     00.02.10                       0                   6,102
        JAIMEC       06.33.10     01.05.44     00.06.31                       0                  67,044
        SCOTTF       06.32.50     00.19.24     00.02.20                       0                   5,782
        BRUCELD2     06.32.32     06.32.19     01.14.18                  11,328                 583,028
        DNSAW        06.16.39     00.16.30     00.01.36                       0                   3,512
        E319         06.15.37     00.26.08     00.04.24                       0                   8,063
        CHUCKP       06.11.43     00.08.01     00.01.40                       0                   4,222
        CICSRAPS     06.01.20     05.59.27     00.06.42                       0                  49,072
        PETERC       05.46.21     00.07.17     00.01.37                       0                   1,855
                     --------     --------     --------     -------------------     -------------------
TOTAL REPORT A      492.51.17    230.59.30     03.51.00                  14,383                 936,810
```

```
                                 RANKING REPORT B - SORTED BY CPU TIME                          07 AUG 1994 PAGE     2

                          ELAPSED       ACTIVE          CPU
             JOB             TIME         TIME         TIME                   TAPE                   DISK
             NAME        HH.MM.SS     HH.MM.SS     HH.MM.SS               IO COUNT               IO COUNT
           --------      --------     --------     --------    -------------------    -------------------

           BRUCELD2      06.32.32     06.32.19     01.14.18                 11,328                583,028
           CHOMS        149.55.50     38.15.43     01.06.52                      0                  1,248
           BRUCELD3      01.16.48     01.16.40     00.48.14                      0                 14,132
           DSSBKUP       01.37.10     01.36.22     00.47.33                 33,800                 60,852
           DICKH         03.29.01     00.42.34     00.30.50                      0                  2,126
           FK4X100       03.33.36     03.31.55     00.27.59                 29,100                236,792
           JSPDLY3       01.00.48     00.57.32     00.27.07                      0                 78,654
           QACOB2        00.35.36     00.31.42     00.16.10                      0                 14,182
           BRUCEAPY      00.37.08     00.35.02     00.12.08                      0                 74,300
           ACSSPDBC      66.19.00     66.18.58     00.11.51                      0                 17,188
           FOCUS2XX      00.27.14     00.26.16     00.11.21                      0                 18,510
           DANK          07.37.52     00.51.38     00.11.09                      0                  8,214
           BRUCEBBB      00.34.46     00.32.58     00.10.57                      0                 74,184
           RAPSQDS       00.30.50     00.29.05     00.10.08                      0                 55,024
           AAPGLU19      02.14.40     02.13.20     00.09.05                      0                203,036
           FRANKIE       07.02.18     00.44.10     00.08.21                      0                 72,889
           AAPGLJ02      00.59.18     00.50.13     00.08.02                      0                 73,610
           MDBACKUP      01.09.04     00.38.43     00.07.48                 15,182                 30,252
           VSAMBKPD      00.19.24     00.19.16     00.07.21                      0                 56,000
           BRUCEPTF      00.27.48     00.22.17     00.06.58                      0                 42,436
           CICSRAPS      06.01.20     05.59.27     00.06.42                      0                 49,072
           DON           08.20.57     00.35.01     00.06.37                      0                 46,874
           JAIMEC        06.33.10     01.05.44     00.06.31                      0                 67,044
           BRUCE         09.24.13     00.20.15     00.05.53                      0                 12,753
                         --------     --------     --------    -------------------    -------------------
   TOTAL REPORT B       286.40.23    135.47.10     07.59.55                 89,410              1,892,400
```

JARSWZ06 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ 'S' OR 'T' AND STEP JOB IND EQ (' 1' OR '11')
SORT
BREAK JOB NAME
COMPUTE ELAPSED/TIME EQ (ELAPSED/TIME/HRS * 3600)
COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/HRS * 3600)
COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600)
RANK 24, ELAPSED TIME, CPU TIME
LIST JOB NAME, ELAPSED/TIME(HS), ACTIVE/TIME(HS),
     CPU/TIME(HS), TAPE/IO COUNT, DISK/IO COUNT
```

JARSWZ06 Execution Description

The INPUT, SELECT, and COMPUTE statements function as in previous examples. Other statements function as follows:

SORT

When no fields are given in a SORT statement, the sort is done on the fields listed in the BREAK statement.

BREAK

BREAK usually specifies a single break field. Only records for the lowest level control break are written to the ranking file.

RANK

The maximum number of records to list in each report and the fields to use in ranking are specified. This product produces a separate report for each field.

LIST

LIST indicates what data fields to print, as usual. In addition, its location tells CA JARS Wizard how many reports to print:

- If the LIST statement appears before the RANK statement, CA JARS Wizard generates the regular detail or summary report for the run in addition to ranking reports.
- If the LIST statement appears after the RANK statement, only the ranking reports are generated.

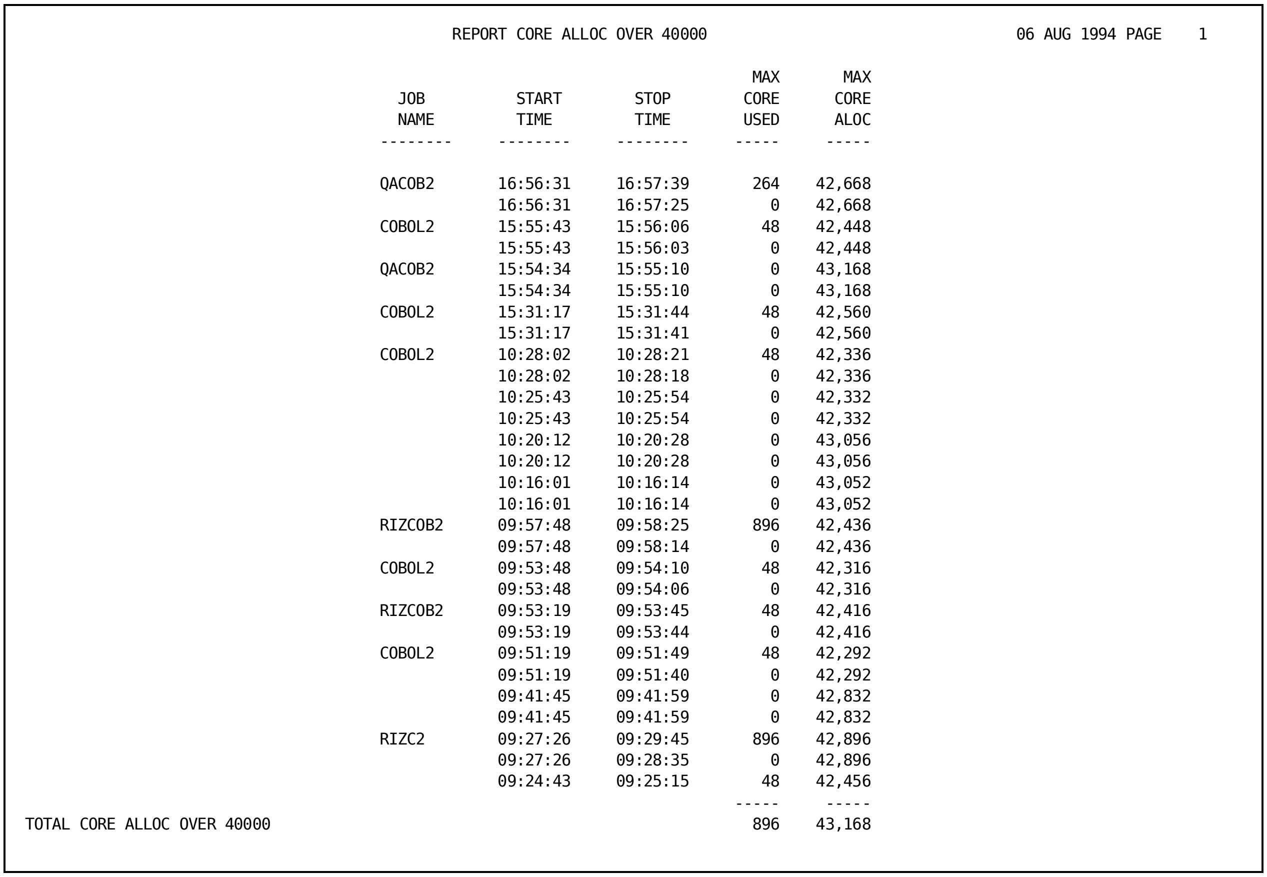
Special Execution Considerations for JARSWZ06

DDname CAIJWOP (an interim/output file) is required to run CA JARS Wizard using JARSWZ06 control statements. See the "Introduction (see page 13)" chapter for more information on the CA JARS Wizard DD statements.

JARSWZ07: Exception Reports

Two exception reports are produced in JARSWZ07. These exception reports can be used to identify jobs with allocated core over 40000 and jobs with core used over 3000.

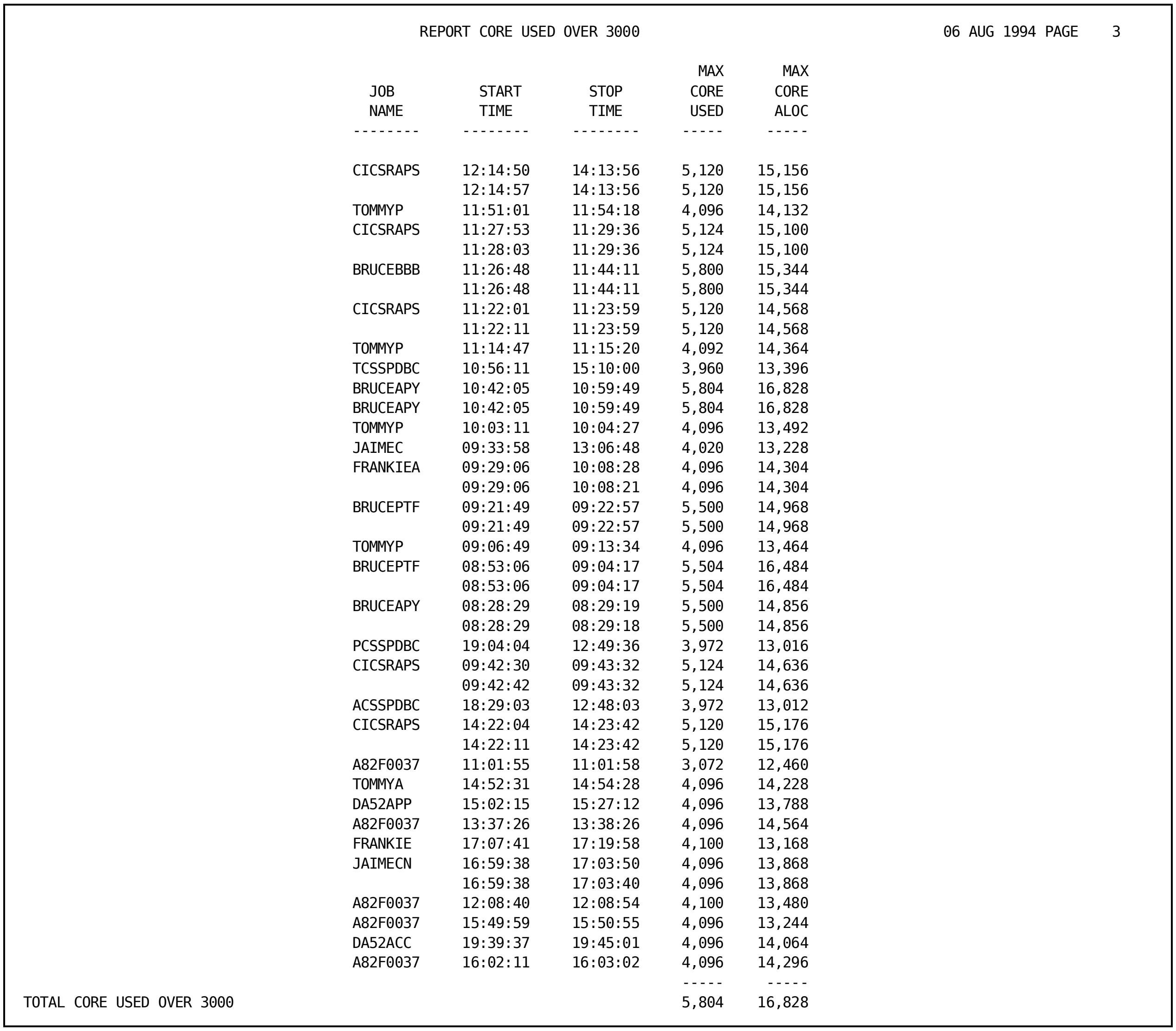
The purpose of exception reporting is to generate short reports of detail records that meet specified conditions.



```
                                        REPORT CORE ALLOC OVER 40000                              06 AUG 1994 PAGE    1

                                                                          MAX       MAX
                                  JOB          START         STOP         CORE      CORE
                                  NAME         TIME          TIME         USED      ALOC
                                 --------     --------      --------      -----     -----

                                 QACOB2       16:56:31      16:57:39        264    42,668
                                              16:56:31      16:57:25          0    42,668
                                 COBOL2       15:55:43      15:56:06         48    42,448
                                              15:55:43      15:56:03          0    42,448
                                 QACOB2       15:54:34      15:55:10          0    43,168
                                              15:54:34      15:55:10          0    43,168
                                 COBOL2       15:31:17      15:31:44         48    42,560
                                              15:31:17      15:31:41          0    42,560
                                 COBOL2       10:28:02      10:28:21         48    42,336
                                              10:28:02      10:28:18          0    42,336
                                              10:25:43      10:25:54          0    42,332
                                              10:25:43      10:25:54          0    42,332
                                              10:20:12      10:20:28          0    43,056
                                              10:20:12      10:20:28          0    43,056
                                              10:16:01      10:16:14          0    43,052
                                              10:16:01      10:16:14          0    43,052
                                 RIZCOB2      09:57:48      09:58:25        896    42,436
                                              09:57:48      09:58:14          0    42,436
                                 COBOL2       09:53:48      09:54:10         48    42,316
                                              09:53:48      09:54:06          0    42,316
                                 RIZCOB2      09:53:19      09:53:45         48    42,416
                                              09:53:19      09:53:44          0    42,416
                                 COBOL2       09:51:19      09:51:49         48    42,292
                                              09:51:19      09:51:40          0    42,292
                                              09:41:45      09:41:59          0    42,832
                                              09:41:45      09:41:59          0    42,832
                                 RIZC2        09:27:26      09:29:45        896    42,896
                                              09:27:26      09:28:35          0    42,896
                                              09:24:43      09:25:15         48    42,456
                                                                          -----     -----
TOTAL CORE ALLOC OVER 40000                                                 896    43,168
```



```
                                        REPORT CORE USED OVER 3000                                  06 AUG 1994 PAGE    3

                                                                         MAX       MAX
                                  JOB          START         STOP        CORE      CORE
                                  NAME         TIME          TIME        USED      ALOC
                                --------     --------      --------     -----     -----

                                CICSRAPS     12:14:50      14:13:56     5,120    15,156
                                             12:14:57      14:13:56     5,120    15,156
                                TOMMYP       11:51:01      11:54:18     4,096    14,132
                                CICSRAPS     11:27:53      11:29:36     5,124    15,100
                                             11:28:03      11:29:36     5,124    15,100
                                BRUCEBBB     11:26:48      11:44:11     5,800    15,344
                                             11:26:48      11:44:11     5,800    15,344
                                CICSRAPS     11:22:01      11:23:59     5,120    14,568
                                             11:22:11      11:23:59     5,120    14,568
                                TOMMYP       11:14:47      11:15:20     4,092    14,364
                                TCSSPDBC     10:56:11      15:10:00     3,960    13,396
                                BRUCEAPY     10:42:05      10:59:49     5,804    16,828
                                BRUCEAPY     10:42:05      10:59:49     5,804    16,828
                                TOMMYP       10:03:11      10:04:27     4,096    13,492
                                JAIMEC       09:33:58      13:06:48     4,020    13,228
                                FRANKIEA     09:29:06      10:08:28     4,096    14,304
                                             09:29:06      10:08:21     4,096    14,304
                                BRUCEPTF     09:21:49      09:22:57     5,500    14,968
                                             09:21:49      09:22:57     5,500    14,968
                                TOMMYP       09:06:49      09:13:34     4,096    13,464
                                BRUCEPTF     08:53:06      09:04:17     5,504    16,484
                                             08:53:06      09:04:17     5,504    16,484
                                BRUCEAPY     08:28:29      08:29:19     5,500    14,856
                                             08:28:29      08:29:18     5,500    14,856
                                PCSSPDBC     19:04:04      12:49:36     3,972    13,016
                                CICSRAPS     09:42:30      09:43:32     5,124    14,636
                                             09:42:42      09:43:32     5,124    14,636
                                ACSSPDBC     18:29:03      12:48:03     3,972    13,012
                                CICSRAPS     14:22:04      14:23:42     5,120    15,176
                                             14:22:11      14:23:42     5,120    15,176
                                A82F0037     11:01:55      11:01:58     3,072    12,460
                                TOMMYA       14:52:31      14:54:28     4,096    14,228
                                DA52APP      15:02:15      15:27:12     4,096    13,788
                                A82F0037     13:37:26      13:38:26     4,096    14,564
                                FRANKIE      17:07:41      17:19:58     4,100    13,168
                                JAIMECN      16:59:38      17:03:50     4,096    13,868
                                             16:59:38      17:03:40     4,096    13,868
                                A82F0037     12:08:40      12:08:54     4,100    13,480
                                A82F0037     15:49:59      15:50:55     4,096    13,244
                                DA52ACC      19:39:37      19:45:01     4,096    14,064
                                A82F0037     16:02:11      16:03:02     4,096    14,296
                                                                        -----     -----
TOTAL CORE USED OVER 3000                                               5,804    16,828
```

JARSWZ07 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ 'S' OR 'T' AND STEP JOB IND EQ (' 1' OR '11')
EXCEPTION IF CORE/USED GT 3000 THEN SORTKEY = CORE USED OVER 3000
EXCEPTION IF CORE/ALOC GT 40000 THEN SORTKEY = CORE ALLOC OVER 40000
COPY JSTRTIME
COPY JSTOPTME
LIST JOB NAME(NR), START/TIME, STOP/TIME, CORE/USED, CORE/ALOC
```

JARSWZ07 Execution Description

The INPUT and SELECT statements function as described in previous examples. The other statements function as follows:

EXCEPTION

The conditions to be reported on are identified. SORTKEY = is a keyword that must be followed by the name of the report. A separate report is generated for each condition.

COPY

Two COPY statements are executed. These statements call in copybooks created and stored in the CAIJWCP data set. The JSTRTIME and JSTOPTME copybooks are provided for you by CA JARS.

LIST

The location of the LIST statement is important. If it follows the EXCEPTION statement, CA JARS Wizard produces only the exception reports. If it precedes the EXCEPTION statement, a detail report is generated in addition to the exception listings. The detail report may be very large unless the request includes selection criteria to reduce the number of records processed.

The (NR) suffix appended to the JOB/NAME field in this example suppresses the printing of the job name on a given line, if the value of JOB/NAME is the same as it was in the previous line.

Special Execution Considerations for JARSWZ07

DDname CAIJWOP (an interim/output file) is required to run CA JARS Wizard using JARSWZ07 control statements. See the "Introduction (see page 13)" chapter for more information on the CA JARS Wizard DD statements.

JARSWZ08: Calendar Report

JARSWZ08 produces a calendar report that lists totals for each day in the month for the number of readers, printers, tapes, disks, other devices, and all devices. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are used in the calculations.

Calendar reports are special reports used to highlight day-by-day trends. Summary totals are produced for each day for up to eight numeric fields.

```
CA                                                        DETAIL REPORT                                CA JARS Wizard PAGE    1
CA JARS Wizard 9502JW000

                 I---------------------------------------------------------------------------------------------------------------I
                 I                                                                                                               I
                 I                                        A U G U S T             1 9 9 3                                       I
                 I                                                                                                               I
                 I---------------------------------------------------------------------------------------------------------------I
                 I               I               I               I               I               I               I               I
                 I    SUNDAY     I    MONDAY     I    TUESDAY    I    WEDNESDAY  I    THURSDAY   I    FRIDAY     I    SATURDAY   I
                 I               I               I               I               I               I               I               I
                 I---------------I---------------I---------------I---------------I---------------I---------------I---------------I
                 I       1       I       2       I       3       I       4       I       5       I       6       I       7       I
                 I               I               I               I               I               I               I               I
NO READERS       I               I               I               I               I               I               I               I
NO PRINTERS      I               I               I               I               I               I               I               I
NO TAPES         I               I               I               I               I               I               I               I
NO DISKS         I               I               I               I               I               I               I               I
NO OTHER DEVICES I               I               I               I               I               I               I               I
NO TOTAL DEVICES I               I               I               I               I               I               I               I
                 I---------------I---------------I---------------I---------------I---------------I---------------I---------------I
                 I       8       I       9       I      10       I      11       I      12       I      13       I      14       I
                 I               I               I               I               I               I               I               I
NO READERS       I               I               I               I               I               I               I               I
NO PRINTERS      I               I               I               I               I               I               I               I
NO TAPES         I               I               I               I               I               I               I               I
NO DISKS         I               I               I               I               I               I               I               I
NO OTHER DEVICES I               I               I               I               I               I               I               I
NO TOTAL DEVICES I               I               I               I               I               I               I               I
                 I---------------I---------------I---------------I---------------I---------------I---------------I---------------I
                 I      15       I      16       I      17       I      18       I      19       I      20       I      21       I
                 I               I               I               I               I               I               I               I
NO READERS       I               I               I             0 I             0 I             0 I             0 I             0 I
NO PRINTERS      I               I               I             0 I             0 I             0 I             0 I             0 I
NO TAPES         I               I               I             0 I             0 I             7 I           181 I             3 I
NO DISKS         I               I               I             5 I             4 I            35 I         3,390 I             9 I
NO OTHER DEVICES I               I               I             0 I             0 I             0 I            20 I             0 I
NO TOTAL DEVICES I               I               I             5 I             4 I            42 I         3,591 I            12 I
                 I---------------I---------------I---------------I---------------I---------------I---------------I---------------I
                 I      22       I      23       I      24       I      25       I      26       I      27       I      28       I
                 I               I               I               I               I               I               I               I
NO READERS       I               I               I               I               I               I               I               I
NO PRINTERS      I               I               I               I               I               I               I               I
NO TAPES         I               I               I               I               I               I               I               I
NO DISKS         I               I               I               I               I               I               I               I
NO OTHER DEVICES I               I               I               I               I               I               I               I
NO TOTAL DEVICES I               I               I               I               I               I               I               I
                 I---------------I---------------I---------------I---------------I---------------I---------------I---------------I
                 I      29       I      30       I               I               I               I           TOTAL MONTH         I
                 I               I               I               I               I               I                               I
NO READERS       I               I               I               I               I               I NO READERS                  0 I
NO PRINTERS      I               I               I               I               I               I NO PRINTERS                 0 I
NO TAPES         I               I               I               I               I               I NO TAPES                  191 I
NO DISKS         I               I               I               I               I               I NO DISKS                3,443 I
NO OTHER DEVICES I               I               I               I               I               I NO OTHER DEVICES           20 I
NO TOTAL DEVICES I               I               I               I               I               I NO TOTAL DEVICES        3,654 I
                 I---------------------------------------------------------------------------------------------------------------I
```

JARSWZ08 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
COMPUTE NO/READERS EQ (NO/RD + 0)
COMPUTE NO/PRINTERS EQ (NO/PR + 0)
COMPUTE NO/TAPES EQ (NO/TP + 0)
COMPUTE NO/DISKS EQ (NO/DK + 0)
COMPUTE NO/OTHER DEVICES EQ (NO/OT + 0)
COMPUTE NO/TOTAL DEVICES EQ (ND + 0)
COMPUTE MOVE RUN/DATE/YMD TO CALENDAR/DATE
SORT RUN/DATE/YMD
CALENDAR NO/READERS,
         NO/PRINTERS,
         NO/TAPES,
         NO/DISKS,
         NO/OTHER DEVICES,
         NO/TOTAL DEVICES
```

JARSWZ08 Execution Description

The INPUT and SELECT statements function as in earlier examples. The other statements function as follows:

COMPUTE

The COMPUTE statements create new data fields and assign them the values from the data record. This is done because the new field names have more meaningful names than the glossary names for the same fields. These fields exist for this run only. The following assignments are made:

- NO/READERS is assigned the value of glossary field NO/RD.
- NO/PRINTERS is assigned the value of glossary field NO/PR.
- NO/TAPES is assigned the value of glossary field NO/TP.
- NO/DISKS is assigned the value of glossary field NO/DK.
- NO/OTHER DEVICES is assigned the value of glossary field NO/OT.
- NO/TOTAL DEVICES is assigned the value of glossary field ND.
- RUN/DATE/YMD is assigned to CALENDAR/DATE.

SORT

Data is sorted by RUN/DATE.

CALENDAR

The CALENDAR statement requests a report generated in calendar format and identifies the data fields to be displayed. CA JARS Wizard accumulates summary totals for each day for each field. There is no need to supply information concerning the days of the month; CA JARS Wizard maintains this information in its own libraries.

CA JARS Wizard can produce multiple calendars for a single request:

- When the input file contains data for more than one month, CA JARS Wizard produces a calendar for each month. SELECT and REJECT statements can be used to restrict the reporting periods.

When the request includes a BREAK statement, CA JARS Wizard produces a separate set of calendars for each value in the highest level BREAK field.

Example:

```
SORT JOB/NAME
BREAK JOB/NAME
```

If this example included these statements, a separate calendar would be produced for each job name. (The sort statement is required whenever a request includes a BREAK statement.)

JARSWZ09: Plot Report

JARSWZ09 graphically displays the tape I/O count, disk I/O count, and total I/O count of jobs. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are listed. Data is presented alphabetically by job name.

```
                                             I / O   A C T I V I T Y   A N A L Y S I S                        06 AUG 1994 PAGE    1

          *---------------TAPE IO---------------*  *---------------DISK IO---------------*   *--------------TOTAL IO---------------*
  JOB           TAPE      TAPE IO COUNT(100)             DISK      DISK IO COUNT(100)             TOTAL      TOTAL IO COUNT(100)
  NAME      IO COUNT  0   50  100  150  200  250     IO COUNT  0   50  100  150  200  250      IO COUNT  0   50  100  150  200  250
--------  ----------  +----+----+----+----+----+   ----------  +----+----+----+----+----+    ----------  +----+----+----+----+----+

ADMINASM         300   *                                  496   *                                   796   *
CALIST         2,919   ***                             10,723   ***********                      13,642   **************
CICSOS16       1,620   **                               4,784   *****                             6,404   ******
IEBCOPDT         239   *                                  276   *                                   515   *
LINK               0                                   23,061   ************************         23,061   ************************
SLSSPZAP         167   *                                  202                                       369   *
VLTABLE            0                                       10                                        10
ZAPSCHD            0                                       34                                        34
--------  ----------  +----+----+----+----+----+   ----------  +----+----+----+----+----+    ----------  +----+----+----+----+----+

               5,245   *****                           39,586   *************************+       44,831   *************************+
  GRAND TOTAL
```

JARSWZ09 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
COMPUTE TAPE PLOT EQ (TAPE/IO COUNT / 100)
COMPUTE DISK PLOT EQ (DISK/IO COUNT / 100)
COMPUTE TOTAL PLOT EQ (TOTAL/IO COUNT / 100)
SORT JOB NAME
BREAK JOB NAME
LIST(SUM) JOB NAME, TAPE/IO COUNT, TAPE PLOT 'TAPE IO COUNT(100)'(G1),
     DISK/IO COUNT, DISK PLOT 'DISK IO COUNT(100)'(G2),
     TOTAL/IO COUNT, TOTAL PLOT 'TOTAL IO COUNT(100)'(G3)
PLOT G1 INCREMENT=10, LENGTH=25
PLOT G2 INCREMENT=10, LENGTH=25
PLOT G3 INCREMENT=10, LENGTH=25
TITLE 2-3=TAPE IO
TITLE 4-5=DISK IO
TITLE 6-7=TOTAL IO
HEADING1C I / O   A C T I V I T Y   A N A L Y S I S
```

JARSWZ09 Execution Description

Considerations for the INPUT, SELECT, SORT, BREAK, TITLE, and HEADING statements are the same as for previous examples. The other statements function as follows:

COMPUTE

The COMPUTE statements are used in this report to simplify the data being graphed. Work fields TAPE PLOT, DISK PLOT, and TOTAL PLOT are created. They store the tape, disk, and total I/O counts for a job divided by 100. Consequently, when the work fields are graphed the graphs are in hundreds of I/Os.

LIST

LIST identifies what data fields to print, as shown in previous examples. In addition, this example uses the (Gn) suffix code to invoke the plot feature. The suffixes (G1), (G2), and (G3) after TAPE PLOT, DISK PLOT, and TOTAL PLOT respectively, specify that these fields are to be printed in plot form. The plot format is described by a PLOT statement. CA JARS Wizard relates the (Gn) suffix with any user-defined PLOT statements. In this case, G1, G2, and G3 are defined in PLOT statements. If a Gn is specified in the LIST statement but the no corresponding PLOT statement is given, CA JARS Wizard uses system default specifications.Refer to the description of the PLOT statement below.

PLOT

The PLOT statements describe the format of three different plots: TAPE PLOT, DISK PLOT, and TOTAL PLOT. In this example all PLOT statements indicate that the plot is to be in increments of 10 for a length of 25 increments (total of 250 x 100 I/Os). Other options on a PLOT statement can specify the range and the character to be used for plotting. See the PLOT statement description in the "Control Statements (see page 59)" chapter for further information.

JARSWZ10: File Matching and Table Lookups

JARSWZ10 uses a secondary file to complete the information needed on a report. The report created is similar to the report created in JARSWZ11.

This example uses a secondary file in the following format:

```
Positions  1   : User information code
           2-21: Department name
```

CA JARS Wizard reads a primary input file record, reads the first position of the USER INFO field, finds a matching user information code in the secondary file, and prints the corresponding department name on the report.

```
                                                                                        07 AUG 1994 PAGE    1


                                               FILE MATCHING: COST CENTER ANALYSIS

                                                                       CPU      ACTIVE
                                                           TOTAL       TIME       TIME         TOTAL
                   DEPT NAME           NBR JOBS         IO COUNT    HH.MM.SS   HH.MM.SS         PAGES
               --------------------    --------       ----------    --------   --------    ----------

               UNKNOWN DEPT CODE             88          286,753    01.26.25  124.54.59        22,375
               COMPUTER DEPT                  3              424    00.00.35   00.02.13           299
               PERSONNEL DEPT                 2            5,338    00.02.27   00.27.34         1,964
               OPERATIONS DEPT                3            6,794    00.01.31   00.23.45         1,560
               PROGRAMMING DEPT           1,321        1,183,837    03.44.15  114.42.07        37,374
               MANAGEMENT DEPT              108           78,187    00.13.41   01.55.53           769
               MARKETING DEPT               460        2,035,195    04.58.44   29.08.19        20,925
               ADMINISTRATIVE DEPT          116          354,421    00.27.44   04.16.13           726
               SHIPPING DEPT                  1                9    00.00.07   00.00.17            42
               SYSTEMS DEPT                  17           13,134    00.35.19   01.01.20         3,178
               --------------------    --------       ----------    --------   --------    ----------

  GRAND TOTAL                             2,119        3,964,092    11.30.48  276.52.40        89,212
```

JARSWZ10 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ 'S' OR 'T' AND STEP JOB IND EQ (' 1' OR '11')
DEFINE(K) DEPT NAME(C)(20) EQ BLANKS
SORT USER INFO(1,1)
BREAK USER INFO(1,1)
INPUT FILE KEY=2CC1-1 MATCH = USER INFO(1,1) TABLE = 2CC2-21 LIST
COMPUTE IF MATCHED THEN DEPT NAME EQ 2CC2-21 ELSE
    DEPT NAME EQ 'UNKNOWN DEPT CODE'
COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600)
COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/HRS * 3600)
LIST(SUM) DEPT NAME(NG), COUNT 'NBR JOBS', TOTAL/IO COUNT,
     CPU TIME(HS), ACTIVE/TIME(HS), TOTAL/PAGES
HEADING4C FILE MATCHING: COST CENTER ANALYSIS
```

JARSWZ10 Execution Description

The SELECT, SORT, BREAK, LIST, and HEADING statements function as discussed previously. The other statements function as follows:

INPUT

The first INPUT statement identifies the primary input file.

The second INPUT statement identifies a secondary file. KEY = 2CC1-1 defines the matching key field for each record in the secondary file. In this example, position 1 of the secondary file is to be used as the matching key field.

The MATCH parameter designates the first position of the USER INFO field of the primary input file as its matching key field. Primary and secondary file records are matched based upon their respective key fields.

TABLE indicates that the secondary file is a table file to be read completely at the start of the run. Consequently, the entire file is available for searching during the run. The field composed of positions 2 through 21 in the secondary file contains the department name information.

The notation 2CC2-21 serves two purposes. It:

- identifies the location of a data field in the secondary file.
- serves as a name for the field.

DEFINE

The DEFINE statement creates a work field DEPT NAME for the run and sets its initial value to blanks. The (K) suffix indicates that the field will be maintained by you and will not be reset to blanks when a new record is processed.

COMPUTE

Three COMPUTE statements are given for this sample. The last two statements have been described in earlier reports (see report JARSWZ03 (see page 26)).

The COMPUTE IF MATCHED condition specifies what CA JARS Wizard should do when a matching user information code is found in the secondary file. Here the work field DEPT NAME is assigned the value of the data in positions 2 through 21 of the secondary file. If a match is not made, the DEPT NAME is assigned a value of UNKNOWN DEPT CODE.

Special Execution Considerations for JARSWZ10

DDname CAIJWI2 (secondary input file) is required to run CA JARS Wizard using JARSWZ10 control statements. See the "Introduction (see page 13)" chapter for more information on the CA JARS Wizard DD statements.

JARSWZ11: Decode Function

JARSWZ11 uses the CA JARS Wizard decode function to create a report identical to the report created in JARSWZ10. A listing of data by cost center is created. This listing gives the department names of each cost center.

Decoding is an alternate way of producing the report that was created in JARSWZ10. In this case, the first position of the USER INFO field is matched with a value in a DECODE table that is built into the program.

```
                                                                                  07 AUG 1994 PAGE    1


                                               C O S T   C E N T E R   A N A L Y S I S

                                                                         CPU       ACTIVE
                                                             TOTAL       TIME        TIME         TOTAL
                      DEPT NAME          NBR JOBS         IO COUNT   HH.MM.SS    HH.MM.SS         PAGES
                   --------------------  --------       ----------   --------    --------    ----------

                   UNKNOWN DEPT CODE           88          286,753   01.26.25   124.54.59        22,375
                   COMPUTER DEPT                3              424   00.00.35    00.02.13           299
                   PERSONNEL DEPT               2            5,338   00.02.27    00.27.34         1,964
                   OPERATIONS DEPT              3            6,794   00.01.31    00.23.45         1,560
                   PROGRAMMING DEPT         1,321        1,183,837   03.44.15   114.42.07        37,374
                   MANAGEMENT DEPT            108           78,187   00.13.41    01.55.53           769
                   MARKETING DEPT             460        2,035,195   04.58.44    29.08.19        20,925
                   ADMINISTRATIVE DEPT        116          354,421   00.27.44    04.16.13           726
                   SHIPPING DEPT                1                9   00.00.07    00.00.17            42
                   SYSTEMS DEPT                17           13,134   00.35.19    01.01.20         3,178
                   --------------------  --------       ----------   --------    --------    ----------

 GRAND TOTAL                                2,119        3,964,092   11.30.48   276.52.40        89,212
```

JARSWZ11 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ 'S' OR 'T' AND STEP JOB IND EQ (' 1' OR '11')
DEFINE(K) DEPT NAME(C)(20) EQ BLANKS
SORT USER INFO(1,1)
BREAK USER INFO(1,1)
DECODE USER/INFO(1,1) TO DEPT NAME,
5 = ADMINISTRATIVE DEPT,
C = COMPUTER DEPT,
3 = MANAGEMENT DEPT,
4 = MARKETING DEPT,
0 = OPERATIONS DEPT,
J = PERSONNEL DEPT,
1 = PROGRAMMING DEPT,
7 = SHIPPING DEPT,
9 = SYSTEMS DEPT,
ELSE UNKNOWN DEPT CODE
COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600)
COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/HRS * 3600)
LIST(SUM) DEPT NAME(NG), COUNT 'NBR JOBS', TOTAL/IO COUNT,
     CPU TIME(HS), ACTIVE/TIME(HS), TOTAL/PAGES
HEADING4C C O S T   C E N T E R   A N A L Y S I S
```

JARSWZ11 Execution Description

All of these statements, except DECODE, function as described in previous examples.

DECODE

The DECODE statement assigns values to the work field DEPT NAME that was created by the DEFINE(K) statement earlier. CA JARS Wizard examines the first position of the USER INFO field in each input record and looks for a match in the DECODE table. If there is one, it assigns the corresponding department name to the DEPT NAME field. If there is no match, it gives DEPT NAME a value of UNKNOWN DEPT CODE.

JARSWZ12: Input File Update

JARSWZ12 produces an output file that contains the same records as the primary input file but with some modifications. A secondary file is used to extract the department name for each primary input file record matching the selection criteria. This department name is then included in the selected records in the output file.

This example is a typical input file update application that processes the primary input file, edits the USER INFO field contents when the record is a job or one-step job record with a processing ID of S (SMF) or T (TSO). An updated file is then created.

```
CA                                                     EXECUTION STATISTICS                        CA JARS Wizard PAGE    1
CA JARS Wizard 9502JW000

 CAJW200I     2168 SORTED INPUT          14.41.47 START
 CAJW202I     2168 OUTPUT RECORDS        14.41.56 END
 CAJW205I     2168 PRE-SORT RECORDS           .09 DURATION
 CAJW206I     2119 SELECTED RECORDS
 CAJW207I       49 SELECT FAILURES
 CAJW211I     2119 MATCHED RECORDS
 CAJW212I     2119 COMPUTATIONS
 CAJW213I       10 FILE 2 RCDS READ
 CAJW214I     2119 FILE 2 MATCHED
 CAJW215I       49 FILE 2 MISSING
 CAJW219I    38888 ADDTL CORE USED
```

JARSWZ12 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ 'S' OR 'T' AND STEP JOB IND EQ (' 1' OR '11')
DEFINE(K) USER/FIELD(C)(16) EQ BLANKS
SORT USER INFO(1,1)
INPUT FILE KEY=2CC1-1 MATCH = USER INFO(1,1) TABLE = 2CC2-17 LIST
COMPUTE IF MATCHED THEN USER/FIELD EQ 2CC2-17 ELSE
    USER/FIELD EQ 'UNKNOWN DEPT'
OUTPUT ALL
```

JARSWZ12 Execution Description

The INPUT statement identifies the glossary to be used with the primary input file.

The second INPUT statement, as discussed in JARSWZ10, is used to define a secondary file for file matching. This is done in conjunction with the COMPUTE IF MATCHED statement. In this example, the secondary file records are formatted to contain a user information code in position 1 and corresponding department name (USER/FIELD) in positions 2 to 17.

The COMPUTE IF MATCHED statement uses the information in positions 2 to 17 to replace the information in the USER/FIELD of the primary input record when a match occurs between a primary input file record and a secondary file record.

Note that department names are only assigned to primary input file for records meeting the selection criteria specified in the SELECT statement; the SELECT statement restricts the COMPUTE IF MATCHED statement processing. If COMPUTE(S) IF MATCHED had been used, however, all primary input file records with a match in the secondary file would be assigned a department name. The (S) suffix code ensures that the COMPUTE is performed before any records are selected.

The new statement used in this example is:

OUTPUT ALL

> This statement specifies that CA JARS Wizard create an output file in the same format as the input file and write out all the records from the input file. Hence, all the input file records, whether selected or not and updated or not, are written out to the output file.

CA JARS Wizard produces a list of execution statistics at the end of all reports. These statistics concern the processing activities. Listed next are some of the execution statistics produced by CA JARS Wizard as displayed in JARSWZ12:

2168 SORTED INPUT

> The number of sorted input records read from the primary input file.

2168 OUTPUT RECORDS

> The number of records written out to the output file is given. Note that the number of output records is equal to the number of input records, as effected by the OUTPUT ALL statement.

2168 PRE-SORT RECORDS

> The total number of records read from the input file.

2119 SELECTED RECORDS

> The number of records selected for processing. (This number represents the number of records meeting the selection criteria.) Note that even though only 2119 records were processed through the COMPUTE IF MATCHED logic, all 2168 input records were written out.

49 SELECT FAILURES

> The number of records from the input file that failed the selection criteria.

2119 MATCHED RECORDS

> The number of input records that had a match with a secondary file record.

2119 COMPUTATIONS

Lists the number of instructions executed in a COMPUTE statement.

In this case, we only have one instruction in a COMPUTE statement. Consequently 2119 COMPUTES were executed on the 2119 selected records. **Note:** However, that this statistic describes the number of records for which the COMPUTE statement was executed. Suppose the COMPUTE statement had been as follows:

```
COMPUTE IF MATCHED THEN
USER/FIELD 2CC2-17,
USER/ID EQ BLANKS
```

Because there are two instructions in this COMPUTE statement, the number of COMPUTATIONS listed in the execution statistics would be doubled.

10 FILE 2 RCDS READ

The number of records read from the secondary file. In this example, these are the table records.

2119 FILE 2 MATCHED

The number of matches for the primary input file against the secondary input file.

49 FILE 2 MISSING

The number of misses for the primary input file against the secondary input file.

38888 ADDTL CORE USED

Lists the number of bytes of virtual storage used.

Special Execution Considerations for JARSWZ12

The following ddnames are required to run CA JARS Wizard using the JARSWZ12 control statements:

- CAIJWI2 - secondary input file
- CAIJWOP - output file

JARSWZ13: Extract File Creation

JARSWZ13 produces an output file containing only selected records from the input file. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are processed.

```
CA                                                     EXECUTION STATISTICS                           CA JARS Wizard PAGE    1
CA JARS Wizard 9502JW000

 CAJW200I     1090 SORTED INPUT           14.53.32 START
 CAJW202I     1090 OUTPUT RECORDS         14.53.37 END
 CAJW205I     2168 PRE-SORT RECORDS            .05 DURATION
 CAJW206I     1090 SELECTED RECORDS
 CAJW207I     1078 SELECT FAILURES
 CAJW219I    37120 ADDTL CORE USED
```

JARSWZ13 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
OUTPUT
```

JARSWZ13 Execution Description

The INPUT, SELECT, and SORT statements are used as in previous examples. The output file creation is specified by the following statement:

OUTPUT

Indicates that an output file is to be created in the same format as the input file, but that it should consist only of the selected records.

The execution statistics produced for this example shows that there were 1090 OUTPUT RECORDS, which is equal to the number of SELECTED RECORDS and less than 2168, which is the number of pre-SORT input records.

Special Execution Considerations for JARSWZ13

DDname CAIJWOP is required to run CA JARS Wizard using the JARSWZ13 sample control statements.

JARSWZ14: Summary File Creation

A summary output file is created in example 14. Only job or one-step job records with processing IDs of S (SMF) or T (TSO) are processed. Summary records are created for each job and are stored in the output file.

```
CA                                                      EXECUTION STATISTICS                        CA JARS Wizard PAGE    1
CA JARS Wizard 9502JW000

 CAJW200I     1090 SORTED INPUT          12.03.25 START
 CAJW202I      406 OUTPUT RECORDS        12.03.33 END
 CAJW205I     2168 PRE-SORT RECORDS           .08 DURATION
 CAJW206I     1090 SELECTED RECORDS
 CAJW207I     1078 SELECT FAILURES
 CAJW216I      406 LEVEL 1 BREAKS
 CAJW219I    42368 ADDTL CORE USED
```

JARSWZ14 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
BREAK JOB/NAME
OUTPUT(SUM)
```

JARSWZ14 Execution Description

The INPUT, SELECT, SORT, and BREAK statements are used as in previous examples. The summary file is created with the following statement:

OUTPUT(SUM)

> OUTPUT(SUM) indicates that CA JARS Wizard should create one summary output record for each lowest level control break. The record contains the accumulated totals for the control break in the numeric fields. Other fields contain values from the first record in the group.

The execution statistics produced for this example tell us that there were 406 OUTPUT RECORDS, which is equal to the number of occurrences of LEVEL 1 BREAKS.

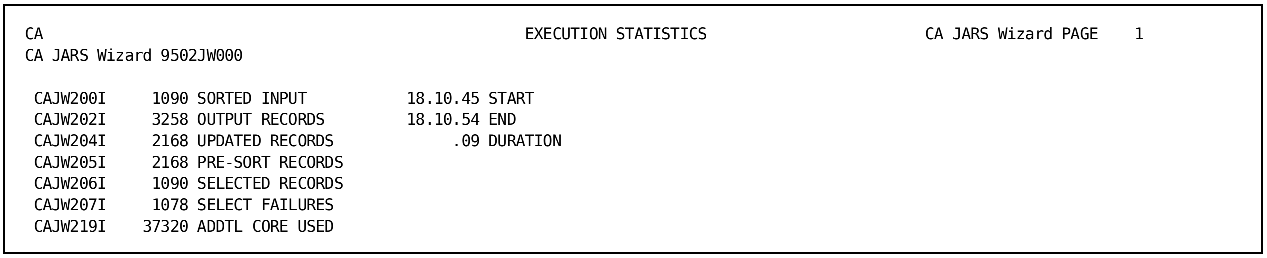
Special Execution Considerations for JARSWZ14

DDname CAIJWOP is required to run CA JARS Wizard using the JARSWZ14 sample control statements. See the "Introduction (see page 13)" chapter for a complete description of CA JARS Wizard DD statements.

JARSWZ15: Output Merge Function

JARSWZ15 produces an output file containing the contents of two input files. The contents of the two input files are **merged** into a single output file.

Although only one input file is listed in the control statements, a secondary file for concatenation, ddname CAIJWOL, is specified in the JCL for the run.



```
CA                                                       EXECUTION STATISTICS                          CA JARS Wizard PAGE    1
CA JARS Wizard 9502JW000

 CAJW200I     1090 SORTED INPUT          18.10.45 START
 CAJW202I     3258 OUTPUT RECORDS        18.10.54 END
 CAJW204I     2168 UPDATED RECORDS            .09 DURATION
 CAJW205I     2168 PRE-SORT RECORDS
 CAJW206I     1090 SELECTED RECORDS
 CAJW207I     1078 SELECT FAILURES
 CAJW219I    37320 ADDTL CORE USED
```

JARSWZ15 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
OUTPUT(UPD)
```

JARSWZ15 Execution Description

The primary input file to be processed is defined by the INPUT statement and all processing statements are performed only against this file.

The file to be updated with new records from the primary input file is written out to the output file without modification. Records from the primary input file are selected, processed, and then written out to the output file.

The merge function is specified by the following statements:

OUTPUT(UPD)

OUTPUT(UPD) indicates that CA JARS Wizard should read the file that is to be updated (defined by DD statement CAIJWOL) and write it to the output file without any processing or modification.

The primary input file records that have been selected and processed are then written out at the end of the output file.

The execution statistics produced for this example tells us that there were 3258 OUTPUT RECORDS.

Note: This is the total of 2168, which is the number of UPDATED RECORDS (or input records) from the file to be updated, and 1090, which is the number of SELECTED RECORDS from the primary input file.

Special Execution Considerations for JARSWZ15

The following ddnames are required to run CA JARS Wizard using the JARSWZ15 control statements:

- CAIJWOL - secondary file for concatenation
- CAIJWOP - output file

JARSWZ16: New File Creation

JARSWZ16 creates a new output file containing summary records for each job.

```
CA                                                     EXECUTION STATISTICS                          CA JARS Wizard PAGE    1
CA JARS Wizard 9502JW000

 CAJW200I     1090 SORTED INPUT          18.05.02 START
 CAJW202I      406 OUTPUT RECORDS        18.05.16 END
 CAJW205I     2168 PRE-SORT RECORDS           .14 DURATION
 CAJW206I     1090 SELECTED RECORDS
 CAJW207I     1078 SELECT FAILURES
 CAJW216I      406 LEVEL 1 BREAKS
 CAJW217I       58 SORT RCD LENGTH
 CAJW219I    39952 ADDTL CORE USED
                             FORMAT OF NEW RECORD
                   POSITION BYTES FORMAT       DESCR
               001   1-  8     8  CHAR         JOB/NAME
               002   9- 11     3  CHAR         PER/GRP
               003  12- 13     2  CHAR         CL
               004  14- 19     6  PACKED       SERVICE/UNITS
               005  20- 23     4  PACKED       CORE/USED
               006  24- 29     6  PACKED       STD LINES/PRNTD
               007  30- 36     7  PACKED       DISK/IO COUNT
               008  37- 38     2  PACKED       NO/DK
               009  39- 40     2  PACKED       NO/TP
```

JARSWZ16 Control Statements

```
INPUT CAWGJR70
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME
BREAK JOB/NAME
OUTPUT(SUM) FILE ITEMS = JOB/NAME,
                         PER/GRP,
                         CL,
                         SERVICE/UNITS,
                         CORE/USED(MX),
                         STD LINES/PRNTD,
                         DISK/IO/COUNT,
                         NO/DK,
                         NO/TP
```

JARSWZ16 Execution Description

The INPUT, SELECT, SORT, and BREAK statements are used as in previous examples. Creation of a new file is specified by the following:

OUTPUT(SUM)

As described earlier, this statement indicates that CA JARS Wizard should create an output record at the lowest control level break.

The ITEMS = parameter identifies which fields from the input records comprise the new output records. Thus, in this example, the output records have the following format:

Positions CC1-8	JOB/NAME
Positions CC9-11	PER/GRP
Positions CC12-13	CL
Positions CC14-19	SERVICE/UNITS
Positions CC20-23	CORE/USED
Positions CC24-29	STD LINES/PRNTD
Positions CC30-36	DISK/IO COUNT
Positions CC37-38	NO/DK
Positions CC39-40	NO/TP

The output fields have the same attributes as the input fields and any time fields (there are none in this example) are 4 bytes binary.

The execution statistics produced for this example tell us that there were 406 OUTPUT RECORDS, this is equal to the number of LEVEL 1 BREAKS.

Special Execution Considerations for JARSWZ16

DDname CAIJWOP is required to run CA JARS Wizard using the JARSWZ16 sample control statements.

Other Sample Reports for CA JARS

The following table list the CA JARS sample reports created using CA JARS Wizard and the members in which you can find the control statements for each report.These members are stored in the CAJROPTN library.

Member Name	Description
JARSWZ17	Job Statistics with Vertical Totals
JARSWZ18	Execution Statistics Only
JARSWZ30	Abnormal End Profile (Programmer Name)
JARSWZ31	Average TSO Session Analysis
JARSWZ32	Class Structure Analysis
JARSWZ33	Detail Charge Audit
JARSWZ34	Job Resource Utilization
JARSWZ35	MVS Performance Group Profile
JARSWZ36	MVS Performance Group Summary
JARSWZ37	MVS Service Requirements by Job Name
JARSWZ38	Monthly System Utilization Calendar
JARSWZ39	Resource Consumption Summary by Day
JARSWZ40	TSO Session Analysis
JARSWZ41	Utilization Summary by Program

Sample Reports for CA JARS DSA Option

CA JARS DSA Option sample reports are created using the CA JARS Wizard glossary CAWGDX60 for files created by CA JARS DSA Option 6.0. The following table lists the CA JARS DSA Option sample reports created using CA JARS Wizard and the members in which you can find the control statements for each report. These members are stored in the CAJROPTN library.

Member Name	Description	Input File
CAIJAWS1	Account Summary Report. Summarizes the charges generated by account code and by category within account code.This report is produced for cycle billing.	&INDEX.SUMOUT(0)
CAIJAWS2	Charge Summary. Summarizes charges by category within account.	&INDEX.SUMOUT(0)
CAIJAWS3	Track Occupancy Summary Report.Summarizes track occupancy by category within account.	&INDEX.SUMOUT(0)
CAIJAWS4	DSNACCT Table.Creates first draft of DSNACCT table.Run just after Cycle Initial.	&INDEX.PTDMAST(0)
CAIJAWD1	Account Detail (single account).Reports on master file detail for a single account number.	&INDEX.PTDMAST(0)
CAIJAWD2	Volume Detail (single volume).Reports on master file detail for a single DASD volume.	&INDEX.PTDMAST(0)
CAIJAWD3	SMS storage class by volume.	&INDEX.PTDMAST(0)
CAIJAWS1	CA JARS DSA Option Account Summary Report.	&INDEX.SUMOUT(0)
CAIJAWS2	CA JARS DSA Option Account Summary Report sorted by charge.	&INDEX.SUMOUT(0)
CAIJAWS3	CA JARS DSA Option Account Summary Report sorted by allocated tracks.	&INDEX.SUMOUT(0)

Member Name	Description	Input File
CAIJAWS4	DSNACCT Table.	&INDEX.PTDMAST(0)

Chapter 3: Control Statements

CA JARS Wizard uses the following control statements. With the exception of a few instances, the exact sequence the control statements are specified in is not important. The instances where it is important are explicitly identified during the discussion of the specific control statements in this chapter. The general rule is to place the control statements that establish an item's identity, such as the INPUT or DEFINE statements, before the item is referenced.

Statement	Function
BREAK	Organizes records into control break groups and calculates group totals for numeric fields.
CALENDAR	Produces a report in monthly calendar format.
CALL	Passes control to a subroutine.
COMPUTE	Inserts arithmetic, assignment, and logical processing instructions into the run and transfers control to a statement other than the next sequential statement in the program.
COPY	Copies a set of one or more control statements or sequential files from the CA JARS Wizard source statement library and inserts them into the run request.
DECODE	Assigns a value to a data field based on the data contents of another field.
DEFINE	Establishes new work data fields and redefines or renames existing data fields.
EXCEPTION	Produces a listing of records meeting special selection criteria.
HEADING	Supplies from one to six lines of text to be printed at the top of each report page.
INPUT	Identifies the input data file or files for the run.
LIST	Specifies the data fields to include in a report print line.
OUTPUT	Specifies the output file to be created.
PARAMETER	Overrides some of automatic page formatting functions, and establishes other options for the run and for compilation output.
PLOT	Produces a report in graph form.
RANK	Produces a summary report with records listed in ascending or descending order of a selected data field.

Statement	Function
REJECT	Specifies which input records to bypass.
SELECT	Specifies which input records to process.
SORT	Defines the sequence in which the input file is to be sorted for the run.
TITLE	Prints a special line of text over two or more adjacent data columns to highlight a relationship.
TITLE(V)	Provides a title for a column of vertical totals.
XTAB	Produces a cross-tabulation report.
*	An asterisk in position 1 designates a comment.

This section contains the following topics:

Syntax

The nine general rules applying to all CA JARS Wizard control statements are described next.

1. A statement can consist of three parts: a statement name, one or more keyword parameters, and a list of one or more data field names. For example:

   ```
   LIST JOB NAME, STEP NAME, ACTIVE TIME
   ```

   The statement name (LIST in the sample statement) can start in any position; however, it must be the first entry in the control statement.

   Keyword parameters can appear anywhere after the statement name except in the middle of a list.

   Lists such as the string of data field names given above, can be of various lengths. Each entry except the last must be followed by a comma.

2. CA JARS Wizard makes extensive use of suffix codes to specify optional special processing. For example, COMPUTE(T) and LIST(SUM) are special variations of the COMPUTE and LIST statements. Similarly, WK35(.4) establishes a data field with four decimal places, and ELAPS TIME(MS) converts time data from seconds to minutes and seconds for printing in a report. Suffixes must be appended to the operation code or the field name without intervening blanks.

3. CA JARS Wizard ignores any entries after position 72 in a control record. Position 72 itself can be used only to indicate that a statement is being continued on the next control record.

4. Blank spaces are important.

   Data field names can contain blank spaces, but never more than one at a time.

   Example:

   ```
   JOB NAME  - valid
   JOB  NAME - invalid
   ```

   Any number of blank spaces can be inserted before or after a data field name, statement name, or punctuation.

5. Character literals should normally be enclosed in single quotation marks (' ') to avoid overhead and confusion with data field names. The quote marks are absolutely required only when the literal contains special characters or spaces.

   Both of the following statements are valid:

   ```
   SELECT APPL CODE EQ AB137596
   SELECT APPL CODE EQ 'AB137596'
   ```

The first statement, with no quotation marks, causes CA JARS Wizard to perform the following checks:

- It looks for a new data field defined with the name AB137596.
- If there is no new data field with that name, it checks the input file glossary for a data field named AB137596.
- If the file search is unsuccessful, it assumes that AB137596 is a data value in the APPL CODE field.

6. Comments

   Any control record with an * in column 1 is treated as a comment. The text is included in the listing for the run, but it has no effect on processing. One or more comments can be inserted at any point in the control statements.

7. Continuation of Control Statements

   Any statement can be continued onto one or more additional records by two methods:

   - Insert any character in position 72.
   - CA JARS Wizard assumes that a statement is continued on the next record if the last entry in the current record is any one of the following:

     ```
     , comma          AND      EQ    EL
                      OR       NE    LE
                      THEN     GT    GE
                      ELSE     LT
                      ALWAYS   EG
     ```

     CA JARS Wizard assumes that position 1 of the continuation record immediately follows position 71 of the record that is being continued. All the rules for the use of blank spaces apply; therefore, if a data field name, literal value, or keyword is interrupted, the text must resume in position 1. Otherwise, initial columns can be left blank if it helps to format the statement listing.

   In addition, COMPUTE, SELECT, REJECT, and EXCEPTION statements are assumed to be continued if the statement only contains the keyword and if the statement ends with an operator (for example, +,-,*,/).

8. Reserved Words

   Certain words have special meaning to CA JARS Wizard and must not be used for anything else. For example:

   ```
   BLANK  BLANKS
   SPACE  SPACES
   ```

   All refer to a field of blank spaces. It can be any length. These words can be used to move a field of blanks to a data field, to compare a field with a field of blanks, or to base selection on whether a field contains blanks.

9. No punctuation is required at the end of a control statement. A period is permitted. The rest of the record must be blank through position 72.

BREAK Statement

Introduction

A BREAK statement indicates when to print a line of group totals.

Format

```
BREAK { field-name1          } ['title[=anyfield]'][(-)][(D)]
         { IF conditional-test1 }

            [(Sn)][(R)][(SB)]
            [(SK)]

         [  { field-name1          } ['title[=anyfield]'][(-)][(D)]
         [  { IF conditional-test1 }

            [(Sn)][(R)][(SB)] ] ... [NOSEQ]
            [(SK)]            ]
```

Use of Parentheses

When you use parentheses, the same operations shown above can be specified in a single instruction:

```
COMPUTE PROC CHG(.2) = (CPU TIME * .016) + (ACTIVE TIME * .008)
```

CA JARS Wizard permits two levels of parentheses in a COMPUTE instruction; that is, it is valid to have one parenthetical operation within another parenthetical operation.

Order of Execution

Multiple operations in a single instruction are executed from left to right in the following priority sequence:

- operations within parentheses
- multiplications
- divisions
- additions
- subtractions

Example:

Consider the following formula and the instruction to execute it for each record:

```
CPU TIME
WORK FIELD = -----------------------      * 100
              CPU TIME + OVRHD TIME
COMPUTE WORK FIELD(.2) = (CPU TIME / (CPU TIME + OVRHD TIME)) * 100
```

CA JARS Wizard carries out this instruction as though it was written as several instructions in the following sequence:

```
COMPUTE WORK FIELD(.2)  =  CPU TIME + OVRHD TIME,
        WORK FIELD  =  CPU TIME / WORK FIELD,
        WORK FIELD  =  WORK FIELD * 100
```

Primary File

```
{CARDS      }           [         {U  }  ]
  INPUT  {file-name  }   [VOL=n] [RECFM =  {F   }  ] [NORWD] [SWLR]
         {FILE       }           [         {FB }  ]
         {member-name}           [         {V  }  ]
         {member-name}           [         {VB }  ]
                                 [         {VBS}  ]

[      ]
       [R=n]        [ B=n  ]
                    [ BF=n ]
                    [      ]
```

Secondary File

```
   INPUT {CARDS      } KEY=fCCa-b MATCH= {fCCa-b    }
         {FILE       }                   {field-name}
         {member-name}                   {          }

[                           ]       [              ]
[TABLE = fCCa-b[,fCCa-b]...] [LIST][RECFM=  {U  } ]
[                           ]       [        {F  } ]
                                    [        {FB } ]
[R=n] [B=n] [UNL]                   [        {V  } ]
                                    [        {VB } ]
      [BF=n]                        [        {VBS} ]
                                    [              ]
```

Parameters

field-name1

Enter one to eight data field names in descending order of importance. All must be glossary names for data fields in the input file or work fields generated before the SORT. The order of entry must be consistent with the SORT statement for the run. The field specified must not be more than 30 characters in length.

IF conditional-test

designates a conditional break. The break occurs only when the condition is satisfied.

See the discussion on the SELECT statement for the format and usage of conditional tests.

All parameters specifying a SELECT condition criterion can be used in the BREAK statement prefixed by IF.

Examples:

```
1. BREAK IF MATCHED
2. BREAK IF DAY EQ '15'
```

title

represents any string of characters to be printed as the heading on the totals line for this data field.

Overrides the automatic break title of *field-name TOTAL* at control break lines.

=anyfield

CA JARS Wizard prints the contents of the specified data field as part of the heading. The value is taken from the first record in the break group. It is printed immediately after the heading text. If there is no heading text, it is printed at the extreme left side of the totals line.

Overrides the automatic break title of *field-name TOTAL* at control break lines.

NOSEQ

ignores errors in the sort sequence for any of the BREAK control fields.

suffix codes

The following suffix codes can be used with data field names in a BREAK statement for detail reports:

(-)

prints a line of dashes after the totals line.

(D)

indicates that the relevant field has been sorted in descending sequence.

(R)

forces the break title for the control break to be right-justified, that is, to the immediate left of the leftmost break total.

(SB)

suppresses the printing of totals for this break, if there is only 1 break at the lower level, or in the case of the lower level break, only 1 record. For LIST(SUM), SB on the lowest level break is ignored.

(Sn)

skips n lines after the total line and before printing the next total line or the first record of the next control group.

(SK)

skips to a new page after printing the totals line for this control group.

If neither (Sn) nor (SK) is specified, the CA JARS Wizard action depends on whether it is generating a detail report (LIST) or a summary report (LIST(SUM)), as follows:

BREAK level	LIST	LIST(SUM)
1	S1	No space
2	S2	S1
3	S3	S2

BREAK level	LIST	LIST(SUM)
4	S4	S3
5	SK	S4
6-8	SK	SK

A suffix specification for a low control break overrides a default specification for a higher control break. For example, if SK is specified for a level 1 break, it applies for all higher breaks.

Sample Usage

Example 1

```
SORT CHARGE CODE, JOB NAME, STEP NAME
BREAK CHARGE CODE, JOB NAME
LIST CHARGE CODE, JOB NAME, STEP NAME, ELAPS TIME
```

These statements produce a report like the following:

```
NETWORK     ORIGIN     RECORD
                         DATE        ID          LOC ID     COUNT
                     ------------    -------    -------     ------
31 AUG 1994     ZNET        CPU-B        126
                     31 AUG 1994     ZNET        CPU-A        967
                                                          -------
NETWORK ID TOTAL     31 AUG 1994     ZNET                   1,093
31 AUG 1994     YARSCS1     CPU-A        878
                     31 AUG 1994     YARSCS1     CPU-A      1,089
                     31 AUG 1994     YARSCS1     CPU-A      1,403
                                                          -------
NETWORK ID TOTAL     31 AUG 1994     YARSCS1                3,370
DATE TOTAL           31 AUG 1994                            4,463
```

Without the BREAK statement, the report would be an uninterrupted listing of step records. The only totals would be the grand totals for the run at the end of the report.

Example 2 (User-Defined Break Titles):

Same as Example 1 except for the following statement:

```
BREAK DATE, NETWORK/ID'TOTAL FOR NETWORK = NETWORK/ID'
BREAK CHARGE CODE, JOB NAME'TOTAL FOR JOB = JOB NAME'
```

With this BREAK statement, the three line headings in the sample report would be as follows:

```
TOTAL FOR JOB ARC200
TOTAL FOR JOBJTR200
DATE TOTAL
```

CA JARS Wizard uses the break title specifications for the JOB NAME heading. There is no break title specification for CHARGE CODE, so the automatic heading is supplied.

Example 3:

```
BREAK SHIFT, DATE, DAY(SK), IF WEEKDAY EQ SUNDAY, DIV CODE(S3)
```

Note: The right most field-name on the break statement is the first level break.

This example causes the fifth level control break to occur when the SHIFT value changes, a third level control break (with a skip to a new page) when the DAY changes, and a second level control break when the WEEKDAY value is SUNDAY. The first level control break occurs for each new DIV CODE, regardless of other break criteria. Three (3) blank lines separate the DIV CODE break line from the next report line.

Considerations

- CA JARS Wizard prepares break total lines automatically. They contain the following information:
  - A heading at the left end of the break total line identifies the break level. For example:

    ```
    JOB NAME TOTAL
    ```

  - The break total line contains group totals for all numeric fields, except those specified in the glossary or user-defined fields with the suffix (NT). Total for some fields are meaningless (such as the TIME field).
  - Character fields specified in the BREAK statement prints for lower level breaks; however, all other character fields will be blank, unless defined by the suffixes (CT), (DF), (DL), or (K).
- You can override page and report formatting at control break lines by using break heading literals or the suffixes discussed in the "Appendix B (see page 155)" chapter.

CALENDAR Statement

Introduction

A CALENDAR statement causes CA JARS Wizard to generate a report in calendar format.

For more information:

Sample Usage (see page 19)

Format

The format is as follows:

```
CALENDAR [yy,] field-name1[, field-name2,..., field-name8]
```

Parameters

yy

Enter two digits for a year. For example, enter 81 for 1981. This field is optional. If it is omitted, CA JARS Wizard uses the year in the first record that it processes.

field-name

Enter from one to eight data field names. The calendar displays totals for the month. The fields must be either packed, binary, or numeric.

Displayed values do not reflect the results of any computations on the control break totals. They are based on automatic accumulations.

Considerations

CA JARS Wizard produces a calendar for each month whenever the highest control break occurs. It is based on the contents of the CALENDAR DATE keyword field, or on the default date field for the glossary if CALENDAR DATE is not set up. If records in the highest break group span a period of four months, CA JARS Wizard generates four calendars whenever it processes the highest level control break. It ignores records for any year other than the one specified.

CALL Statement

Introduction

The CALL statement transfers program control to a specified CA JARS Wizard program.

Note: Avoid using CALL statements in the same run as a COMPUTE GO TO EOF statement.

Format

```
CALL[(I)] program-name
    [(P)]
```

Options

CALL

The CALL statement without a suffix is executed after the SORT and after all COMPUTE and SELECT statements.

CALL(P)

The (P) suffix causes the CALL routine to execute before the SORT, and immediately before any COMPUTE(P) statements.

CALL(I)

The (I) suffix causes the CALL routine to execute immediately after reading the input record.

Parameters

program-name

Specify the name of the cataloged program to invoke for special CA JARS Wizard processing.

Note: See the *CA JARS Installation Guide* for a list of these special programs.

Considerations

This is **not** a user-exit facility. Due to the complicated parameter control procedures, no attempt should be made to call programs other than those supported and documented as part of the relevant product.

COMPUTE Statement

Introduction

The COMPUTE statement inserts arithmetic instructions and conditional logic into the CA JARS Wizard run. These instructions are executed for each record as it is processed.

COMPUTE instructions can be inserted at various points in the processing cycle. Several suffix codes are used to specify these points.

Format

```
[label] COMPUTE[(S)]  { Assignment instructions            }
               [(P)]  { IF, THEN, ELSE, ALWAYS instructions }
               [(U)]  { MOVE instructions                  }
               [(n)]  { GO TO instructions                 }
               [(T)]  { NP command                         }
               [(G)]
```

Options

label

is a one to eight character label used in conjunction with the GO TO option of the COMPUTE statement.

COMPUTE

is executed after processing all the SELECT statements and after the SORT, if there is one.

COMPUTE(S)

is executed on all input records before any SELECT, REJECT, or CALL statements are processed, except CALL(I). Any new fields generated at this time can be used as a basis for the SORT and BREAK statements.

COMPUTE(P)

is executed on all selected input records by the SELECT(P) or SELECT statement positioned before the COMPUTE(P) statement. Any new fields generated at this time can be used as a basis for the SORT and the BREAK statements.

COMPUTE(U)

is executed on selected records by the SELECT statement and after performing the SORT, but before the execution of the SELECT(S) statement.

COMPUTE(n)

is executed on control break totals, prior to any other control break processing. Specify n as a number between one and eight, inclusive. n indicates the control break level, where 1 is the lowest level control break in the run. Compute(n) on standard deviation items does not affect the value carried to level n+1.

Example:

COMPUTE(3) instructions are executed only during processing for the third-level control break. They have no effect on first and second level totals. They may alter totals for the third and all higher levels.

COMPUTE(T)

is executed on control breaks after all other processing, but before printing. COMPUTE(T) instructions are executed during every control break at all levels, including grand totals.

COMPUTE(G)

is executed on the grand totals level, before any other processing. At this point, the entire report is written, except for printing of grand totals.

Note: The count keyword cannot be used to list the total number of jobs.

Example:

```
DEFINE TOTAL/JOBS(N)(4) =0
COMPUTE IF STEP JOB IND EQ (' 1' OR '11') THEN
        TOTAL/JOBS EQ  TOTAL/JOBS + 1
```

Assignment Instructions

Assignment instructions move the value of the operand on the right-hand side of the equals symbol (EQ or =) to the left-hand side operand.

Format

```
field-name1[(x,y)][(suffixes)]  { =  }  { field-name2[(x,y)]    }
                                { EQ }  { value                 }
                                        { fCCa-b                }
                                        { arithmetic-expression }
```

Operands

field-name1

can be either a new field, a glossary-defined field, or a field previously defined with a DEFINE or COMPUTE statement.

If it is a new field name, it can contain up to 16 characters. Blank characters are valid, but can only be used one at a time. Otherwise, the valid characters are the letters A through Z and the digits 0 through 9. The slash (/) character can also be used; it causes the characters of the name before it and after it to print as two separate lines on a column heading.

CA JARS Wizard establishes the field as a four-byte packed decimal field unless the field name has a suffix code specifying a different format.

If it is a new field, suffixes can be used to specify special attributes for field-name1. (Refer to Appendix B (see page 155) for a discussion on the suffixes.)

field-name2

must be a glossary-defined field or a field previously defined with a DEFINE, DECODE, or COMPUTE statement.

Field-name1 and field-name2 must have the same or similar format. Allowable combinations of formats are:

character and numeric
packed and numeric
packed and binary
binary and numeric

Field Format:	**Can be converted to:**
binary	numeric and packed formats
character	numeric format
numeric	binary, character, and packed formats
packed	binary and numeric formats

[(x,y)]

specifies a subfield within a character field, where x is the starting position and y is the length of the subfield. This is not valid for date or numeric fields.

value

can be any character or numeric literal. A character field can contain up to 255 letters, digits, and special characters. A numeric field can be up to 9,999,999 unless the field name has a suffix code establishing a larger format.

The keywords BLANK, BLANKS, SPACE, and SPACES can be used to establish values of blanks for character fields.

The keywords LOVALUE and LOVALUES can be used to establish values of zeros for numeric fields, according to the field's format.

fCCa-b

The CC notation refers to positions in a nonglossary file in the run.

a

specifies the starting position of the field in the record.

b

specifies the ending position of the field in the record.

f

specifies which physical file is being referenced, that is, f = 1 specifies the first physical file, etc. The f can be omitted if only one nonglossary file is present.

Arithmetic Expressions

Format

```
field-name1[(suffixes)] =
     field-name2[(suffixes)]  { + } field-name3[(suffixes)]
                              { - }
                              { * }
                              { / }
             [  { + } field-name4[(suffixes)] ]...
             [  { - }                         ]
             [  { * }                         ]
             [  { / }                         ]
```

The following arithmetic operations can be performed:

- + addition
- - subtraction
- * multiplication
- / division

Example:

A run may include the following COMPUTE instructions to calculate billing costs:

```
COMPUTE CPU CHG(.2) = CPU TIME * .016,
        ACTIVE CHG(.2) = ACTIVE TIME * .008,
        PROC CHG(.2) = CPU CHG + ACTIVE CHG
```

These sample instructions call for two multiplications and an addition. The results are saved in CA JARS Wizard work fields.

Use of Parentheses

When you use parentheses, the same operations shown above can be specified in a single instruction:

```
COMPUTE PROC CHG(.2) = (CPU TIME * .016) + (ACTIVE TIME * .008)
```

CA JARS Wizard permits two levels of parentheses in a COMPUTE instruction; that is, it is valid to have one parenthetical operation within another parenthetical operation.

Order of Execution

Multiple operations in a single instruction are executed from left to right in the following priority sequence:

- operations within parentheses
- multiplications
- divisions
- additions
- subtractions

Example:

Consider the following formula and the instruction to execute it for each record:

```
                    CPU TIME
WORK FIELD = -----------------------  * 100
              CPU TIME + OVRHD TIME
COMPUTE WORK FIELD(.2) = (CPU TIME / (CPU TIME + OVRHD TIME)) * 100
```

CA JARS Wizard carries out this instruction as though it was written as several instructions in the following sequence:

```
COMPUTE WORK FIELD(.2)  =  CPU TIME + OVRHD TIME,
         WORK FIELD  =  CPU TIME / WORK FIELD,
        WORK FIELD  =  WORK FIELD * 100
```

GO TO Instructions

There are five branching instructions: GO TO END, GO TO EOJ, GO TO EOF, GO TO NEXT, and GO TO Label. They cause CA JARS Wizard to skip past one or more COMPUTE instructions of the same level.

Note: Avoid using a COMPUTE GO TO EOF statement in the same run as a CALL statement.

GO TO END

Execution of all COMPUTE instructions at the same level is terminated. For example, COMPUTE(P)..., GO TO END terminates execution of COMPUTE(P) instructions, but has no effect on COMPUTE instructions where the statements have a different suffix.

GO TO NEXT

Program control branches to the next COMPUTE statement at the same level. For example:

```
COMPUTE(P) ....,
           IF CODE EQ 1 THEN GO TO NEXT,
           ELSE...
           .
           .
           .
COMPUTE (P)...,
```

GO TO NEXT terminates execution of the first COMPUTE(P) group and branches to the group of statements belonging to the next COMPUTE(P) statement. If there are no more COMPUTE statements at the same level, CA JARS Wizard resumes normal processing.

GO TO Label

Program control branches to a labeled COMPUTE statement at the same level. The statement must be further down in the same program. Backward branches are invalid.

The label can be any name from one to eight characters in length. It must be used only to identify COMPUTE statements that are being given control through a GO TO instruction.

GO TO EOF

CA JARS Wizard is forced to terminate input processing and to start the next process in its execution. Avoid using a COMPUTE GO TO EOF statement in the same run as a CALL statement.

GO TO EOJ

Execution of CA JARS Wizard is terminated in an orderly manner. No further computation occurs and all files are closed.

Considerations

COMPUTE statements with the same suffix are considered to be at the same level and executed at the appropriate processing logic time. (See the "Appendix A" chapter for an overview discussion of the CA JARS Wizard program logic.)

The COMPUTE statements of the same level are executed in the physical sequence that they appear among the control statements.

COMPUTE IF, THEN, and ELSE Instructions

IF, THEN, and ELSE instructions are inserted into a run in the same way as arithmetic instructions.

Example:

```
COMPUTE FIELD ONE = ELAPS TIME = ACTIVE TIME,
COMPUTE IF FIELD ONE EQ 300 THEN FIELD TWO(C)(1) = '*'
```

This example is a computation. If FIELD ONE is equal to 300 seconds, CA JARS Wizard inserts an asterisk into a flag field to highlight the record when it is printed in a report.

ALWAYS Instructions

An IF instruction can be followed by any number of other instructions to be executed if the condition is met. The list ends when CA JARS Wizard encounters an ELSE instruction, an ALWAYS instruction, or another IF.

Example:

```
IF condition THEN instruction-1,
                  instruction-2,
                  instruction-3,
ELSE              instruction-4,
                  instruction-5,
ALWAYS            instruction-6,
                  instruction-7
```

When a record meets the IF condition, instructions 1, 2, 3, 6, and 7 are executed. When the condition fails, instructions 4, 5, 6, and 7 are executed.

NO Nesting

IF and ELSE conditions cannot be nested. A new IF statement starts fresh. For example:

```
IF condition-a THEN  instruction-1,
                     instruction-2,
IF condition-b THEN  instruction-3,
                     instruction-4,
ELSE                 instruction-5,
ALWAYS               instruction-6
```

This set of instructions can have four possible outcomes, as follows:

First IF Satisfied	Second IF Satisfied	Execute Instructions
Yes	Yes	1,2,3,4,6
Yes	No	1,2,5,6
No	Yes	3,4,6
No	No	5,6

THEN, ELSE and ALWAYS

THEN, ELSE and ALWAYS are keywords followed by an assignment, MOVE, or GO TO instruction. NP commands can also be used. The ELSE and ALWAYS sections are optional.

Multiple Conditions

An IF statement can contain up to 60 conditional tests connected by AND or OR.

AND

The THEN instruction is executed only when the conditions immediately preceding and immediately following the AND are satisfied.

OR

The THEN instruction is executed when a condition on either side of the OR is satisfied.

One level of parentheses can be used to group related conditions.

MOVE Instructions

The MOVE instruction is a specialized function that should be used only to convert date field, hexadecimal field, or character field formats. It moves data from field-name-a to field-name-b and converts the format of the data to field-name-b's format. Field-name-a must be a previously defined field, but field-name-b can be a new field.

Data Conversions

The MOVE instruction performs the following conversions:

- Date field conversions

  Fields that are defined with date attribute suffixes can be converted from one internal date format to another. The acceptable CA JARS Wizard internal date formats are:

  YYMMDD (C)(6) or (N)(6) or (P)(4) YY/MM/DD (C)(8) MMDDYY (C)(6) or (N)(6) or (P)(4) MM/DD/YY (C)(8) DDMMYY (C)(6) or (N)(6) or (P)(4) DD/MM/YY (C)(8) YYDDD (C)(5) or (N)(5) or (P)(3) .fo on

- MOVE date-field TO WEEKDAY

  CA JARS Wizard evaluates the date value in the date-field and place in WEEKDAY, a keyword and reserved area, the appropriate day of the week as MONDAY, TUESDAY, WEDNESDAY, etc.

- MOVE date-field TO DAY OF WEEK

  CA JARS Wizard evaluates the date value in the date-field and place in DAY OF WEEK, a keyword and reserved area, the appropriate numeric day of the week, as follows:

  1 for MONDAY,
  2 for TUESDAY, etc.

- MOVE field-name-1 TO field-name-2

  CA JARS Wizard converts the value of the data in field-name-1 to the format of field-name-2. Allowable conversions are:

  hexadecimal to character
  character to hexadecimal

  In a character to hexadecimal conversion, nonhexadecimal characters are treated as zeros.

  If the length of the character field is not equal to twice the length of the hexadecimal field, the resulting field is truncated or padded on the right as appropriate. Character fields are padded with spaces; hexadecimal fields are padded with hexadecimal zeros.

  Note: Subfields of packed, binary and numeric fields are invalid.These formats must be converted to character format first.

NP Command

This command forces a page skip whenever it is executed.

Caution: This command should not be confused with the NP suffix. See the Print Control Suffix Codes section in the "Appendix B (see page 155)" chapter for more details.

Considerations

If this command is to be effective, it must be contained in a COMPUTE statement that executes during the report generation stage. If it is placed in a COMPUTE(P) statement, it is ignored.

Example

```
COMPUTE IF NEW JOB NAME THEN NP
```

A page break is performed whenever a new job name is encountered.

COPY Statement

Introduction

The COPY statement copies a set of one or more CA JARS Wizard control statements from the Source Statement Library. The copied statements replace the COPY statement in the CA JARS Wizard program.

Format

```
COPY name
```

Parameters

The name parameter is the name assigned to the statements when they are cataloged in the Source Statement Library. A CA JARS Wizard program can contain any number of COPY statements.

The COPY facility can be used to copy an entire program or any part of one. Copied statements have a + prefix in the control statement listing for the run, and are clearly indicated in the listing with START and END statements.

Up to 8 levels of nesting is permitted.

DECODE Statement

Introduction

The DECODE statement assigns a value to a data field based on the contents of another data field.

Format

```
DECODE[(P)] field-name-a TO field-name-b[(suffix)],
            field-name-a-value-1 = field-name-b-value-1,
            field-name-a-value-2 = field-name-b-value-2,
            &bul.
            &bul.
            &bul.
            ELSE field-name-b-value-n
```

Options

DECODE

The DECODE statement without a suffix is executed after the SORT, and immediately before any COMPUTE statements.

DECODE(P)

The (P) suffix causes the DECODE statement to execute before the SORT, and immediately before any COMPUTE(P) statements.

Parameters

field-name-a

Enter the name of any file data field or work field in the run. CA JARS Wizard examines the contents of this field as each record is processed. It then looks for a match in the DECODE table and assigns a value to field-name-b. If there is no match, processing continues.

field-name-b

Enter the name of any file data field or work field. If the field was not previously defined, a new data or work field is established. The name can have a suffix code to specify type, format, and length. It can be in character or numeric format, depending on the values specified in the DECODE list that follows. If nothing is specified, CA JARS Wizard assumes that it is four bytes, packed decimal.

The value in field-name-b is either reset to zero or is blank at the beginning of each record processing cycle.

field-name-a-value

List the possible values that can appear in field-name-a. The field can be character or numeric.

field-name-b-value

Specify what value should be assigned to field-name-b when the matching field-name-a value occurs. This value can be in either character or numeric format. If it contains a special character, it should be enclosed in single quote marks.

ELSE

Optionally, specify a default value for field-name-b. This value is assigned when CA JARS Wizard cannot find a match for field-name-a in the DECODE table. If there is no ELSE and there is no match, then field-name-b either remains set to zero, blank, or is set to the initial value of its DEFINE statement.

Sample Usage

```
DECODE USERID TO USER NAME(20)(C),
CMSAED = 'ANNE D.',
CMSCED = 'CRAIG D.',
ELSE UNKNOWN USER
```

In this example a new data field is set up called USER NAME. It is formatted as 20 character bytes and assigns a value to the new field based on the charge code in each input record.

The DECODE statement is a convenient way to specify a number of tests and assignments.

The sample DECODE statement above is equivalent to the following instructions:

```
DEFINE USER NAME(C)(20) = BLANK
COMPUTE IF USERID EQ 'CMSAED' THEN
    USER NAME = 'ANNE D.',
        IF USERID EQ 'CMSCED' THEN
    USER NAME = 'CRAIG D.',
ELSE USER NAME = 'UNKNOWN USER'
```

Note: When a string used in a DECODE statement includes either a comma or period followed by a space, or a space immediately followed by any special character, then the string must be enclosed in single quotes.

Example:

```
DECODE USER/INFO (2,1) TO WORK/FIELD(C) (30)
       A = 'APPLES & ORANGES',
       B = '%BANANAS',
       C = CHERRIES, CANTALOUPE, CUCUMBERS,
       G = 'GRAPES, GRAPEFRUIT, AND GUAVA',
       P = PEARS? PINEAPPLES?,
       ELSE NO. FRUIT
```

CHERRIES, CANTALOUPE, CUCUMBERS, is invalid because of the commas; otherwise, A, B, C, P and ELSE statements are valid.

DEFINE Statement

Introduction

The DEFINE statement performs the following functions:

- assigns a new name to a file data field for this run only
- assigns a name to one portion of a larger data field
- creates a new data field and establish an initial data value

The DEFINE statement must precede any CA JARS Wizard statement that refers to the new name; however, work fields created with the DEFINE statement cannot be used in SORT, BREAK, or other preSORT statements.

Format

```
DEFINE[(K)] field-name1[(suffixes)]  { EQ }   { field-name2[(x,y)] }
                                     { =  }   { value              }
                                     {    }   { fCCa-b             }
```

Options

DEFINE

causes field-name1 to be reset to the value with which it is initialized every time a new record is processed. Synonyms are created by defining field-name1 to be equal to an existing field-name2, keyword, or work item. All numeric fields generated by a DEFINE statement are subject to automatic processing functions.

DEFINE(K)

is generally used to generate a new-field that contains a constant value, or contents that are to be maintained by you. CA JARS Wizard does not reset the field.

Automatic processing functions are not performed on numeric fields generated by a DEFINE(K) statement.

The current value of a character field is printed when generated by a DEFINE(K) statement in a detail or summary line, if the field is specified in the LIST statement, and even if it is not a BREAK field.

Parameters

field-name1

can contain up to 16 characters. Blank spaces are valid, but can only be used one at a time. Otherwise, the only valid characters are the letters A through Z and the digits 0 through 9.

The slash (/) character can also be used; it causes the characters of the name before it and after it to print on two separate lines in a column heading.

CA JARS Wizard establishes the field as a four-byte packed decimal field unless one of the following conditions applies:

- the field name has a suffix code specifying a different format
- the initial value is character format data. CA JARS Wizard establishes the field as character with a length equal to the initial value

Field-name1 can be qualified by using suffixes. The suffixes specify format, length, scaling, editing, and print attributes for the field, overriding the defaults. (See the "Appendix B (see page 155)" chapter for a discussion on the suffixes.)

field-name2

refers to any data field defined in the glossary, a keyword, or a previously defined work item. Once defined, the field can be referenced by its glossary name or its new name. The column heading is based on the name used in the LIST statement.

(x,y)

The (x,y) suffix designates a subfield within an existing character field.

x

designates the first position of the subfield within the existing field.

y

designates the length of the subfield.

For example, (2,4) indicates that only the second, third, fourth, and fifth characters of the existing data field are considered. This suffix cannot be specified for date or numeric fields.

value

can be any character or numeric literal. A character field can contain up to 255 letters, digits, and special characters. A numeric field can be up to 9,999,999 unless the field name has a suffix code establishing a larger format.

The reserved words BLANK, BLANKS, SPACE, and SPACES can be used to establish initial values of blanks for character fields.

The reserved word LOVALUE can be used to establish initial values of zeros for numeric fields, according to the field's format.

fCCa-b

The CC notation refers to positions in a nonglossary file in the run. CC stands for Control Column.

f

specifies the physical file (1 = first physical file, etc.). This can be omitted if only one nonglossary file is present.

a

specifies the starting position of the field in the record.

b

specifies the ending position of the field in the record.

Sample Usage

Example 1

```
DEFINE APPL CODE(1,3) EQ SYSTEM CODE
```

This statement makes it possible to treat the first three characters of the application code as a unique data field named SYSTEM CODE.

Example 2

```
DEFINE FIELD ONE EQ 'GEORGE WASHINGTON'
DEFINE FIELD TWO(C)(17) EQ 'GEORGE WASHINGTON'
DEFINE FIELD THREE(.2)  =  0
DEFINE FIELD FOUR(N)(5)(.2)  =  00
```

In this example, FIELD ONE and FIELD TWO are identical. FIELD THREE is set up as a four-byte packed decimal field with two decimal places. FIELD FOUR is a five-position numeric field capable of holding a value up to 999.99.

Example 3

```
DEFINE COUNTER A EQ 0
DEFINE(K) COUNTER B EQ 1
COMPUTE COUNTER A = COUNTER A + 1,
        COUNTER B = B + 1
```

In this example, both counters are incremented by 1 for each new record. Counter A is reset to zero every time a new record is read. Counter B retains its value and could be used to keep track of a running total controlled by you.

Example 4

```
DEFINE DEPARTMENT(C)(12) EQ 2CC17-28
```

This statement creates a new field called DEPARTMENT and assigns it the value of whatever data appears in positions 17 through 28 of a record in the second physical file of the run (a nonglossary file).

EXCEPTION Statement

Introduction

An EXCEPTION statement defines requirements for one or more exception reports. The EXCEPTION statement cannot be used in the same processing run as the RANK, OUTPUT, LIST(BOTH), or PLOT control statements.

Format

```
                    EXCEPTION IF conditional-test1  [AND conditional-test2 ]  ...
                                                    [OR conditional-test2  ]



                                 THEN SORTKEY  { EQ } report-name1
                                               { =  }



                             [ IF conditional-test3 [ AND conditional-test4  ] ...
                             [                      [ OR conditional-test4   ]


                                    THEN SORTKEY { EQ } report-name2  ]  ...
                                                 { =  }               ]
```

Parameters

IF...THEN SORTKEY

IF THEN instructions function in the same way as in the COMPUTE statement. Use AND and OR if necessary to establish multiple conditions. For example:

```
IF OP CODE = 'R' AND DIV CODE = 1915 THEN
  SORTKEY = AUDIT RERUNS
```

This example checks two data fields before accepting a record for inclusion in the AUDIT RERUNS exception report. Refer to the description of the SELECT statement for rules about the syntax of conditional tests. Note that the equal sign (=) and the EQ characters can be used interchangeably.

The THEN keyword must be followed by SORTKEY=report-name.

report-name

contains up to 30 characters that are used for sorting and must be unique for each report.

The name must contain only the letters A through Z and the digits 0 through 9. Special characters are not allowed.

Sample Usage

A sample statement follows:

```
EXCEPTION IF MINIDISK/OWNER EQ 'MIKEC' THEN SORTKEY = MIKEC MINIDISKS,
          IF MINIDISK/OWNER EQ 'PATMIN' THEN SORTKEY = PATMIN MINIDISKS,
          IF DAY GT 30 THEN SORTKEY = LATE RUNS
```

This sample statement produces three reports. The first two list records with the specified minidisk owners. The third shows job steps that ran after the thirteenth of the month, if any.

Considerations

- EXCEPTION, OUTPUT, LIST(BOTH), PLOT, and RANK statements are mutually exclusive in a run. Also, if the input file is unusually long, you may want to suppress the regular report by checking that there is no LIST statement before the EXCEPTION statement. Summary report facilities are not available in an exception report run.
- If there is a LIST statement before the last EXCEPTION statement(s), a detail report is generated in the first pass.

  If there is a LIST statement after the last EXCEPTION statement(s), it is used for the exception reports; otherwise, the LIST statement preceding the EXCEPTION statements is used.
- If you request a detail report prior to the exception report and want it sequenced, you must supply a SORT statement prior to the EXCEPTION statement.

  If you want the exception report to be sequenced, you must supply a SORT statement after the EXCEPTION statement.

  Should you require both a detail and an exception report and want both reports sequenced, you must include a SORT statement both before and after the EXCEPTION statement (that is, you need two SORT statements).
- Exception reports are created using a two-pass process. The fields available for use in the second pass are:
  - CA JARS Wizard keywords
  - All input glossary fields
  - SORTKEY
  - Fields computed in the SECOND pass

  When applying the exception feature, you must recompute the items on the LIST statement for exception reports that are not derived directly from the input record. You must recompute these items as part of the second pass. All statements following an EXCEPTION statement are applied to the second pass.

HEADING Statement

Introduction

HEADING statements print up to six lines of text at the top of each report page. Text can be placed at the left, right, or center of each of the six lines.

Format

```
HEADINGn[L] text[=field-name]
        [R]
        [C]
```

Parameters

n

is a line number, 1-6.

L, R, or C

represent the position of the heading on the line. The default is C (center).

L

left-hand corner

R

right-hand corner

C

center

blank

center is assumed

text

More than one character of heading text must be used in a HEADING statement. The text may not, however, exceed the page width.

= field-name

CA JARS Wizard prints the contents of the specified data field as part of the heading. It is printed immediately after the heading text.

Note: If the field-name contains imbedded blanks, please code slashes instead of blanks. For example, "Program/Name" instead of "Program Name".

Sample Usage

```
HEADING1C JOBS LISTED BY CPU TIME
HEADING3L AUDIT DIVISION
HEADING3R CONFIDENTIAL
HEADING4L NEW YORK OFFICE
```

These sample statements produce the following lines at the top of each page of the report:

```
JOBS LISTED BY CPU TIME
AUDIT DIVISION                                             CONFIDENTIAL
NEW YORK OFFICE
```

INPUT Statement

Introduction

An INPUT statement is required for every CA JARS Wizard run. It identifies the primary file to be processed. An additional INPUT statement is required for each secondary file in the run. The maximum number of INPUT statements is four.

Note: You must specify an INPUT statement for the input glossary prior to referencing any fields defined by that glossary.

Format

The INPUT statement has two formats. One applies to the primary file for a run. The other applies to any secondary files or tables in the run.

Parameters

B=n

describes the block length of the input file records. This is only used when the block length is different from the definition specified for the file in the glossary.

BF=n

describes the blocking factor of the input file records. This is only used when the block length is different from the definition specified for the file in the glossary.

CARDS

indicates that the input file is a card file.

FILE

indicates that the input file is on tape or disk.

filename

must be the name of the glossary for the primary file for the run. The file can be any of the product files that are defined for CA JARS Wizard.

Note: For more information, see the *CA JARS Installation Guide* for a list of file names defined for your system.

KEY=fCCa-b

identifies the location of the key field in the secondary file records. A value of 2CCa-b applies to the second input file (first secondary file); 3CCa-b applies to the third; and 4CCa-b applies to the fourth, if any. The key field must not exceed 64 characters in length.

For example, KEY=2CC1-2 means that the key field is in the first two positions of the second input file records. KEY=3CC1-2 applies to the first two positions of the third input file. Records can be of any size, but a and b must not exceed the record length of the file.

LIST

instructs CA JARS Wizard to display the complete secondary file in the report listing.

MATCH =

identifies the key field in the primary file and other secondary files. A match exists when this field contains the same data value as the KEY field in the secondary file. All files must have the same sort order for their key fields if the table function is not used. Primary input records can be processed regardless of whether they are matched, depending on the criteria in the selection statements.

member-name

identifies the name under which the file is cataloged in the Source Statement Library, CAIJWCP.

NORWD

is required in addition to the VOL parameter when the input is from two or more files on a single tape reel. NORWD suppresses the automatic tape rewind after the first file. For example:

```
INPUT CAJCICS VOL = 3  NORWD
```

This example tells CA JARS Wizard that the primary input file occupies three files on a single tape reel and that the automatic tape rewind is to be suppressed.

Note: This parameter applies to VSE only.

R=n

describes the record length of the input file records. This is only used when the record length is different from the definition specified for the file in the glossary.

RECFM

is required whenever the record format for the file is incorrect. For example, nonglossary files are assumed to have fixed-length, unblocked, 80-byte records.

SWLR

suppresses abnormal termination when the input file block length is not equal to the expected input block size.

Note: This parameter applies to VSE only.

TABLE=fCCa-b

This parameter serves two purposes.

- It identifies a table file. The entire file is read before the start of processing.
- It defines data fields in the secondary file that are retained and thereby referenced. The position specification can be used in other CA JARS Wizard statements as a data field name, or a more meaningful name can be assigned in a DEFINE statement. For example, 3CC15-22 can be used in any statement as the name of the data field in positions 15-22 of the current record from the third input file.

UNL

specifies that the input file resides on an unlabeled magnetic tape.

Note: This parameter applies to VSE only.

VOL=n

is required when an input file occupies two or more files on one or more volumes. A maximum of 27 can be used.

n specifies the number of volumes or files to be processed. Without this, CA JARS Wizard assumes end of input on reaching the end of the first file. (See the "Sample Usage (see page 19)" chapter for the required JCL for multiple files/volumes.)

Note: This parameter applies to VSE only.

The KEY and MATCH parameters apply whenever the run includes a secondary file. They give the information necessary to match records from the two different files. TABLE initiates a binary lookup procedure of data that resides in main storage. TABLE can be applied only when the secondary file is limited and can fit in main storage. When TABLE is used, CA JARS Wizard reads the entire secondary file into memory at the start of the run and sorts it. Without it, the secondary file must somehow be sorted in the same sequence as the primary input file, and matching is performed by reading one record at a time from the primary and secondary files.

Considerations

- CA JARS Wizard normally performs file matching after the SORT step, if there is one. Deferring this action generally improves performance speed. Nevertheless, statements that are executed before the SORT can refer freely to data fields in secondary files. The execution of the file matching step is moved before the SORT so that the secondary records are available in time for the first reference.
- All files must be sorted so that their key fields are in the same sequence except when table files are used as secondary files.
- For more information about matching, see the discussion of the MATCHED and MISSING parameters for the SELECT (see page 110) and COMPUTE (see page 71) statements.
- The fCCa-b names have the same status as other data field names. They can be used in control statements, and all suffix codes apply. For example, 1CC15-18 is assumed to be a four-byte alphanumeric field. 1CC15-18(N)(.2) defines a four-digit numeric field with two decimal places. Input data must conform to this specification.
- If only one nonglossary file is present, then all references to fields in that file can be made using the notation CCa-b, that is, f need not be specified.

Sample Usage

```
INPUT primary-input-file
INPUT FILE KEY=2CC1-1 MATCH = E.RUN/TYPE TABLE = 2CC2-17 LIST
```

where *primary-input-file* identifies the primary file to be processed.

In this example, the secondary file has a run type in position 1. Positions 2 through 17 are also flagged for reference during processing.

CA JARS Wizard reads the entire secondary file at the start of the run. It then reads the primary file, record by record, as usual. It checks the run type field in each record and searches for a match in the table. The program can then refer freely to fields in the table.

```
LIST DATE, JOB NAME, 2CC2-17, ELAPS TIME(HS), E.CPU TIME(HS)
```

The COMPUTE or DEFINE statement can be used to assign a more meaningful name, such as RUN TYPE, and a column heading to the 2CC2-17 data field in the secondary file. The MATCHED and MISSING parameters can be used in SELECT and COMPUTE statements to specify the action that should be taken when CA JARS Wizard finds (MATCHED) or cannot find (MISSING) a matching record in the secondary file.

LIST Statement

Introduction

A LIST statement identifies what data fields to print, where to put them on the page, and whether special editing is needed. LIST(BOTH) cannot be specified in the same CA JARS Wizard processing run as an EXCEPTION, RANK, PLOT, or OUTPUT statement.

Format

```
 [     [    ]]                                    [    ][    ]
  LIST [(SUM)[(T) ]] field-name1['heading1'][(suffixes)][(T) ][(Cn)]
       [     [(Tn)]]                                    [(Tn)][(Sn)]
       [     [(TG)]]                                    [(TG)]
       [           ]                                    [    ]
       [(BOTH)     ]
       [(T)        ]
       [(Tn)       ]
       [(TG)       ]
       [           ]
[(Fcn)] [SPACEn]
[                                    [    ][    ]
      [,field-name2['heading2'][(suffixes)][(T) ][(Cn)][(Fcn)][SPACEn]
      [                                    [(Tn)][(Sn)]              ]
      [                                    [(TG)]                    ]
      [                                    [    ]                    ]
```

Options

LIST

for a detail report.

LIST(SUM)

for a summary report. (Only numeric items appear unless specific suffixes are used.)

LIST(BOTH)

to get a detail report followed by a summary version of the same report. This option can only be used when the run does not perform OUTPUT, RANK, PLOT, or EXCEPTION operations.

Parameters

field-name

Any file data field or work field name can be specified. The maximum number of names is 32. All names must fit on one line; the real limit, however, is the size of the print line. CA JARS Wizard assumes the maximum print line size is 132 positions, unless the run includes a PARAMETER statement with a different WIDTH specification.

Enter the names in the order that the fields should appear in the report, from left to right. CA JARS Wizard automatically inserts blank spaces to separate fields in the report. In the LIST statement, names must be separated by commas.

'heading#'

A data field's name is also used as its column heading when printed in a report. The field can have more than one valid name because of a DEFINE statement in the run. The name in the LIST statement is used as the column heading.

A column heading can be one, two, or three lines in length. Names can be segmented into two or three lines by replacing any spaces in the field name by a slash (/). For example:

TEST FLAG appears as:

```
TEST  FLAG
```

but

TEST/FLAG appears as:

```
TEST
FLAG
```

An alternate column heading is specified by entering its text within single quote marks, immediately after the data field name.

suffix codes

For more information, see Appendix B (see page 155) for a discussion of suffix codes that can be used to specify or override print attributes of field-names.

SPACEn

causes CA JARS Wizard to insert n blanks between two adjacent fields. For vertical totals, use the appropriate suffix code to space n lines.

(T), (Tn), (TG)

These suffix codes create vertical totals. Vertical totals are created for data fields at control breaks, but the fields are not included in either the detail or summary print line. The vertical totals for the fields are printed in column one after the regular line of control break totals.

(T)

Supplies totals for the field at every control break.

(T1)-(T8)

Supplies totals for the field only at specified control breaks. T1 is the lowest control break, and T8 refers to the highest control break.

(TG)

Supplies totals for the field only with grand totals at the end of the report.

These suffix codes can be written following the LIST keyword. In this case, they apply to all items in the statement unless overwritten for an individual item by a different suffix.

The sample report on the next page illustrates the use of vertical totals.

Caution: If you use LIST field-name 'heading' (N,N), you will receive error messages. The correct format is: LIST field-name(N,N) 'heading'

```
CA                                                        DETAIL REPORT                            CA JARS Wizard PAGE    1
CA JARS Wizard 2.3 9710JW000




APPL NAME TOTAL

APPLICATION        ACCOUNTING DEPARTMENT
TRAN CODE                      APQIOD
TOT/TRAN/CNT                     99,982
AVERAGE/RESPONSE/TIME               0.5
AVERAGE/CPU/TIME                  0.027
TOT/DB/CALLS                  1,678,839
TOT/MSG Q/ISRT                  189,389
PGM/ABENDS                            5


APPL NAME TOTAL

APPLICATION        LEGAL DEPARTMENT
TRAN CODE                      LESIOE
TOT/TRAN/CNT                     45,892
AVERAGE/RESPONSE/TIME               1.5
AVERAGE/CPU/TIME                  0.028
TOT/DB/CALLS                    983,789
TOT/MSG Q/ISRT                   38,290
PGM/ABENDS                            9


APPL NAME TOTAL

APPLICATION        UNKNOWN
TRAN CODE                      WHUSHJ
TOT/TRAN/CNT                     87,662
AVERAGE/RESPONSE/TIME               1.3
AVERAGE/CPU/TIME                  0.034
TOT/DB/CALLS                  2,364,125
TOT/MSG Q/ISRT                  362,646
PGM/ABENDS                           25

  GRAND TOTAL

 APPLICATION       UNKNOWN
 TRAN CODE                     WHUSHJ
 TOT/TRAN/CNT                   233,536
 AVERAGE/RESPONSE/TIME              1.0
 AVERAGE/CPU/TIME                  0.03
 TOT/DB/CALLS                 5,026,753
 TOT/MSG Q/ISRT                 590,325
 PGM/ABENDS                          39
```

The statements used to generate the vertical total lines after a change in X.APPL NAME (the control break in this example) are:

```
LIST X.APPL NAME'APPLICATION'(T), X.TRAN/ID'TRAN CODE'(T),
     X.TOT/TRAN/CNT(L8)(T), AVG/RESP'AVERAGE/RESPONSE/TIME'(L4)(T),
     AVG/CPU'AVERAGE/CPU/TIME'(L4)(T), X.TOT/DB/CALLS(L8)(T),
     X.TOT/MSG Q/ISRT(L6)(T), X.PGM/ABENDS(L5)(T)
```

or

```
LIST(T) X.APPL NAME'APPLICATION', X.TRAN/ID'TRAN CODE',
     X.TOT/TRAN/CNT(L8), AVG/RESP'AVERAGE/RESPONSE/TIME'(L4),
     AVG/CPU'AVERAGE/CPU/TIME'(L4), X.TOT/DB/CALLS(L8),
     X.TOT/MSG Q/ISRT(L6), X.PGM/ABENDS(L5)
```

Other Usage Considerations

- In the following example, the text given in single quotes is substituted for the field name as the column heading.

  ```
  LIST JOB NAME, DISK SIO(PC)'DISK SIO PERCENT', ACTIVE TIME
  ```

  A display of this would be as follows:

  ```
            JOB NAME          DISK SIO PERCENT          ACTIVE TIME
  ```

- Vertical totals (created using the (T), (Tn), and (TG) suffixes) are ordinarily printed in column 1, regardless of the report location of the vertical total field. Each vertical total is printed on a separate line.

  To print vertical totals on the same line, use one of the suffixes (Cn) or (Sn). The vertical total is printed in column n, where n has a value from 1 to 6. (A maximum of six vertical totals can be printed on the same line.) The titles of each vertical total line (except the first in the series) are suppressed. If vertical totals are put in the same column on the same line, fanning may occur.

  Specify the fill character for vertical total fanning with the Fcn suffix code. Fill character c is placed at interval n to the left of the item, until a nonblank space is encountered. CA JARS Wizard uses a period (.) as the default fill character and 2 as the default interval.

OUTPUT Statement

Introduction

An OUTPUT statement produces a new data file. CA JARS Wizard creates the new file in addition to supplying any detail reports that have been requested.

An OUTPUT statement is not valid in the same run as the LIST(BOTH), RANK, or EXCEPTION statements.

Format

```
{CARDS        }       [    ]
  OUTPUT[(SUM)][(UPD)]  {FILE          } [R=n] [B=n ]
                        {glossary-name}        [BF=n]
                                               [    ]

[ALL]  [UNLABELED]
[                            ] [        {F   } ]
         [ITEMS = name-1,..., name-n ] [RECFM = {FB } ]
         [ITEMS = LIST               ] [        {V  } ]
         [                           ] [        {VB } ]
                                       [        {VBS} ]
```

Options

OUTPUT

An OUTPUT statement with no suffix codes or parameters creates a copy of all selected records from the primary input file. Data values may be altered by processing during the run.

(SUM)

creates a summary file. The file contains one record for every lowest control break. Numeric fields contain group totals. Other fields contain values from the first record in the group.

When using this suffix with an output glossary that contains different record types, only one of the record types can be selected for output and summing. This can be achieved by either forcing the processing of a single record type from the input file with a SELECT statement, or prefixing the glossary name specified on the OUTPUT statement with a record type.

(UPD)

concatenates two or more input files. First, the file is copied and updated without modification (except that block sizes may be altered). CA JARS Wizard then reads the primary input file, performs any processing requested by control statements in the run, and adds it to the end of the output file.

This facility could be used, for example, when adding a new day's records to a period-to-date history file. First, CA JARS Wizard copies all records from the existing file; then it adds the new day's records, thus performing all daily processing.

Parameters

ALL

The output file contains every record from the input file. Records can be sorted. The run can select certain records for processing. Output records reflect any changes to data values in processed records, but all records are included in the output file, regardless of whether or not they are selected for processing.

B = n

specifies a block size for the output file. n is the number of bytes in each block. If this parameter is omitted, the output block size is either the same as the input block size or is calculated from the blocking factor parameter.

BF = n

specifies the blocking factor, which can be used instead of specifying the block size.

CARDS

causes CA JARS Wizard to write its output file to SYSPCH.

Note: This parameter applies to VSE only.

FILE

causes CA JARS Wizard to write a newly formatted file, whose contents are user-specified, to a file with the name of CAIJWOP.

glossary-name

any of the file glossaries defined for CA JARS Wizard. Output areas are reformatted to correspond to output glossary definition. The rules for the initialization of fields are determined by the field formats. Fields in the output record can be referenced with the O prefix.

Prefix the glossary name with a record type if the glossary is a variable record type glossary and only one record type is to be written to the output file.

ITEMS = LIST

instructs CA JARS Wizard to include all fields that are in the LIST statement for the run.

When using the LIST statement, the OUTPUT statement must follow.

The ITEMS parameter is not valid if an output glossary name is specified or defaulted to the input glossary.

ITEMS = name-1,...

identifies specific data fields to be included in the output file records and the order that they appear. The parameter can identify each field by name.

R = n

specifies the output record length. n is the number of bytes in each record. If omitted, CA JARS Wizard assumes that output records are the same length as the input records, or if an ITEMS= parameter has been specified, then the record length equals the sum of the item lengths. An error condition occurs if new data fields have been added during the run, unless the record length is changed.

RECFM

is required whenever the record format assumed by CA JARS Wizard is incorrect. For example, nonglossary files are assumed to have fixed-length, unblocked, 80-byte records.

UNLABELED

suppresses the writing of tape labels when the output file is assigned to tape.

Note: This parameter applies to VSE only.

In general, if the OUTPUT glossary is not equal to the INPUT glossary, then the following processing occurs:

- The OUTPUT record is initialized to binary zeros.
- Any common record area for the OUTPUT glossary is initialized.
- If the OUTPUT glossary was prefixed by a record type, then the output record is formatted as per that record type. In the case of variable records, an initial RDW is created.
- CA JARS Wizard processing occurs.
- If the OUTPUT glossary was not prefixed, any initialized fields in the output record are now formatted and an RDW set up if not already set by you.
- The record is now written or summarized.

Considerations

- The OUTPUT statement must precede any COMPUTE statement that references O-type output items.
- When running in z/OS or VM, do not code DCB information for output files (unless you are writing to a member of a partitioned data set), as this is provided by CA JARS Wizard.
- If the OUTPUT glossary name is FILE or CARDS, the RECFM parameter is limited to F or FB. V, VB or VBS are reserved for use by CA-developed glossary names.

PARAMETER Statement

Introduction

A PARAMETER statement specifies options for the run. These options override automatic page formatting functions and list options, and specify the compilation output.

Format

```
PARAMETER [AVERAGE] [BETWEEN=n] [CURRENCY=x] [DESC] [DIAG]
[EDIT] [EUROPE] [HOLD] [LEFT] [LINES=n] [MISSING]
[NC] [NG] [NOPC] [NOUNDER] [RELEASE] [RIGHT]
[SB] [SPACE=n] [VZ] [WIDTH=n] [XREF]
[ZERO] [ZT]
```

Options can be entered in any order. No punctuation is required to separate them. There are no required options, and the statement can be omitted when not needed.

Options

AVERAGE

produces a line of average values at each control break. The line of averages appears immediately after the usual line of group totals. Each value is the numeric total for the field in the group divided by the number of records in the group. The average can be suppressed for a given field by giving an (NA) suffix after its name in the LIST statement.

BETWEEN = n

overrides automatic field separation and places n spaces between each field in the report.

CURRENCY=x

specifies a character to be used instead of the normal currency symbol, the dollar sign character ($).

DESC

includes in the XREF listing a description of all glossary items and keywords referenced. This description is up to 40 characters in length, is taken from the glossary, and cannot be modified.This option is ignored for RANK reports during the second pass.

DIAG

prints diagnostic information after the report. This information includes the location of phases that are loaded into storage.

EDIT

indicates that the control statements should be processed for purposes of editing the syntax and program listing; that is, CA JARS Wizard compiles the program and produces the compilation listings and does not proceed to the program execution phase.

EUROPE

prints numeric fields in European notation. Groups of three digits are separated by periods. Decimal values are set off by commas. For example:

U.S.	EUROPE
1,375.96	1.375,96

If nothing is specified, U.S. notation is followed.

HOLD/RELEASE

HOLD suppresses the printing of control statements in the listing for a run. You may want to use this feature when running an error free program on a regular basis. RELEASE terminates a HOLD parameter, so that the statements following it are listed. The set of control statements can include a mixture of PARAMETER HOLD and PARAMETER RELEASE statements. If RELEASE and HOLD are used in the same PARAMETER statement, HOLD overrides RELEASE.

LEFT/RIGHT

LEFT overrides automatic centering of reports, and creates the requested report left-justified on the page. RIGHT also overrides automatic centering of reports, and creates the requested report right-justified on the page. When LEFT and RIGHT are used in the same run, (not necessarily in the same PARAMETER statement), LEFT overrides RIGHT.

LINES=n

specifies a maximum number of print lines per page. If this option is omitted, the installation standard is followed.

MISSING

causes CA JARS Wizard to display all keys for which no match was found when secondary files are processed with the TABLE parameter.

NC

forces CA JARS Wizard to sort the complete input record, together with all extra preSORT work fields, rather than sort only those fields used in the program. No SORT compression is forced when the OUTPUT file is the same glossary type as the INPUT.

NG

suppresses automatic grand total processing.

NOPC

suppresses the error handling routine.

NOUNDER

suppresses automatic underlining of field totals.

SB

suppresses the printing of control break totals for all break levels if there is only 1 break at the immediate lower level, or in the case of the lowest level break, only 1 record. For LIST(SUM), SB on the lowest level break is ignored. This parameter must be specified before any BREAK statement.

SPACE=n

Enter a number from 1 through 9 to indicate the number of lines to space after printing each detail record line. For example, enter SPACE=2 to double-space the report, 3 for triple-spacing, etc.If this option is omitted, CA JARS Wizard assumes single-spacing except after break group total lines.

VZ

prints vertical totals containing zero values.

WIDTH=n

Enter the maximum number of print positions per report line. The maximum is 132, the minimum is 72. If this option is omitted, CA JARS Wizard assumes 132.

XREF

causes CA JARS Wizard to list all data field names that are referenced, and to identify each statement that uses them. Without this option, only a listing of input statements is supplied.

ZERO

suppresses the printing of zero values. Without it, at least one zero is printed when the data value is zero.

ZT

forces the printing of time fields having a value of zero.

PLOT Statement

Introduction

The PLOT statement describes the plot format of a field to be graphed, overriding the plot format default specifications of CA JARS Wizard.

Format

```
PLOT Gn [CHARACTER=x] [INCREMENT=n] [LENGTH=n] [OVERFLOW=x]
[RANGE=x-y] [UNDERFLOW=x]
```

Parameters

Gn

n is a value from 1 to 6, identifying the plot format. There can be up to six formats in a run. Data field names in the LIST statement use a suffix code (G1) to (G6), indicating that their values are to be displayed according to a corresponding PLOT statement format, or according to CA JARS Wizard defaults.

CHARACTER = x

x is the character to be displayed. It can be any letter, digit, or special character. Default is '*'.

INCREMENT = n

n is the increment between print positions. If n is 5 and the data value is 15, then three print positions are required to display it. Data values are divided by the increment value and rounded. Default is range/length or 5, minimum is 1, and maximum is 9,999.

LENGTH = n

n is the number of print positions required for the plot. Default is range/increment or 20, minimum is 5, and maximum is page width minus 2.

OVERFLOW = x

x is the character to be displayed when the data value is greater than the maximum RANGE. It appears one position to the right of the last print position. The default is '+'.

RANGE = x-y

x-y indicates the expected minimum and maximum values of the field to be plotted. Default from minimum is 0, maximum is 999, and to maximum is 9,999. Default is 0 to the increment length or 0 to 100.

UNDERFLOW = x

x is the character to be displayed if the data value is less than the minimum RANGE. This character appears one position to the left of the first print position. The default is '-'.

Sample Usage

For more information, see the "Sample Usage (see page 19)" chapter for a sample of the PLOT Report.

RANK Statement

Introduction

The RANK statement defines requirements for one or more ranked records reports. Records are always ranked at the lowest control break level specified on the BREAK statement. A BREAK statement is required when doing ranking.

A RANK control statement cannot be included in the same run as an EXCEPTION, OUTPUT, LIST(BOTH), or PLOT control statement.

Format

```
[    ]                  [ [                      ]
       RANK [ALL,] field-name1[(A)] [,[ALL,] field-name2[(A)]]...
            [n,  ]                  [ [n,  ]                ]
            [    ]                  [ [    ]                ]
```

Parameters

n

n is the number of groups to be listed. It can be any number from 1 to the maximum possible number of control break level-1 groups. Enter ALL to list all groups. If omitted, CA JARS Wizard assumes 20.

ALL

indicates all groups are ranked.

field-name

represents the name of any file data field or work field. Enter up to ten names. Each entry produces a separate report.

(A)

reverses the sort order for a report. It requests an ascending sort, from A to Z, 0 to 9. Without this suffix, ranking is on descending values, and the listing runs from 9 to 0, Z to A.

Sample Usage

```
RANK 30, CPU TIME
```

This example produces a listing of 30 groups in descending order by CPU time. CA JARS Wizard suppresses the printing of remaining groups for this report.

```
RANK 25, CPU TIME, 10, ACTIVE TIME, PAGES PRINTED, ALL, PAGE FAULTS
```

This example produces four listings. The first shows 25 groups in descending order by CPU time; the second lists ten groups in descending order by active time; the third lists ten groups in order by pages printed; and the final report shows all groups ordered by page fault statistics.

```
RANK 10, CORE SIZE, CORE SIZE(A)
```

The first report in this example shows the ten groups with the largest core size. The second report shows the ten groups with the smallest core size.

Considerations

A RANK statement is invalid in the same run as an EXCEPTION, OUTPUT, LIST(BOTH), or PLOT control statement. These are mutually exclusive functions.

The total amount of storage required for the RANK reports requested in a single step should not exceed 640 KB. This is the product of the total number of lines in the RANK reports and the length of the records passed to the RANK report generation process. If this storage exceeds 640 KB, a warning message will be issued, and the reports will be truncated.

The RANK statement separates statements executed on the first pass from those executed on the second pass. Only fields that are specified on the LIST statement are available in the second pass, so the fields on which the report is to be ranked must be specified in the LIST statement. The only fields that can be referenced in the second pass, other than those specified in the LIST statement are:

- keywords
- work fields computed in the second pass
- REPORT - A single character field used to identify the separate ranking reports
- RANK - the name of the field on which ranking is alone. This can only be used for HEADINGS.

Note: The RANK statement can work on individual records as they are read into CA JARS Wizard and processed before the SORT/BREAK (detail level) or after they go through SORT/BREAK processing (summary level).

To RANK at the detail level, use COMPUTE(P) statements for any calculations on the field(s) to be RANKed.

To RANK at the summary level, use all COMPUTE (but not COMPUTE(P)) statements for any calculations on the fields(s) to be RANKed.

Detail example:

```
COMPUTE(P) 1.CPU/TIME = 1.CPU/TIME / 1000
COMPUTE(P) CPU RATIO(B) (3) (.6) = 1.CPU/TIME / 1.ELAPSED/TIME
SORT
BREAK USERID, DATE, CPU RATIO
RANK 20, CPU RATIO
LIST USERID, DATE, 1.CPU/TIME, 1.ELAPSED/TIME, CPU RATIO(L10)
```

Summary example:

```
SORT
BREAK USERID, DATE, CPU RATIO
COMPUTE 1.CPU/TIME = 1.CPU/TIME / 1000
COMPUTE CPU RATIO(8) (3) (.6) = 1.CPU/TIME / 1.ELAPSED/TIME
RANK 20, CPU RATIO
LIST USERID, DATE, 1.CPU/TIME, 1.ELAPSED/TIME, CPU RATIO(L10)
```

REJECT Statement

Introduction

A REJECT statement indicates which records from the primary input file are to be bypassed. Its rules and format are the same as for the SELECT statement, except that it bypasses records rather than selecting them.

REJECT statements can be used with SELECT statements.

```
[   ] {                  [                     ]   }
      REJECT[(P)] {conditional-test1 [AND conditional-test2]...}
            [(S)] {                  [OR conditional-test2 ]   }
            [(L)] {MATCHEDn          [                     ]   }
                  {MISSINGn                                    }
                  {SAME field-name                             }
                  {NEW field-name                              }
```

Note: For more information, see SELECT (see page 110) statement on the next page for the rules of usage.

SELECT Statement

Introduction

A SELECT statement indicates which records in the primary input file are to be processed. CA JARS Wizard accepts records that meet the SELECT statement conditions. It bypasses all others.

Any number of records can be selected or bypassed in a run. The action has no effect on the contents of the input file itself.

Format

```
 [   ] {                  [                     ]   }
      SELECT[(P)] {conditional-test1 [AND conditional-test2]...}
            [(S)] {                  [OR conditional-test2 ]   }
            [(L)] {ALL               [                     ]   }
            [   ] {MATCHEDn                                    }
                  {MISSINGn                                    }
                  {SAME field-name                             }
                  {NEW field-name                              }
```

Format of Conditional Tests

```
field-name1 {NL       *< }  {field-name2}
                           {NG       *> }  {value      }
                           {EQ       =  }  {keyword    }
                           {NE   or  *= }
                           {GT       >  }
                           {LT       <  }
                           {EG/GE    => }
                           {EL/LE    =< }
```

Options

Suffix codes indicate when SELECT and REJECT criteria are applied.

SELECT

Record selection takes place before or after COMPUTE(P), depending on the physical sequence of the SELECT and COMPUTE(P) statements, and before sorting.

SELECT(S)

Record selection takes place after sorting, and after any COMPUTE(U) statements but before any COMPUTE statements.

SELECT(L)

The criterion applies solely to the determination of which detail records are to be printed by the LIST statement. Other records can be selected and processed in the run, but they are not represented in the listing. This does not apply to summary listings created using LIST(SUM).

SELECT(P)

Record selection takes place before sorting and before COMPUTE(P), irrespective of the physical sequence of the SELECT and COMPUTE(P) statements, but after any COMPUTE(S) statements.

Parameters

field-name

can be any data field name from the input file or a work area generated before the SELECT is executed. It can be qualified to base selection on a subfield within the field, as, for example, ACCOUNT/NUMBER(1,3).

In a conditional test, when two field names are entered, the data in the second field is compared with the data in the first field.

Example:

```
SELECT 6.PASSWORD/COUNT EQ 6.PASSWORD/LIMIT
```

This example selects any record in which the password limit is the same as the password count.

Valid comparison operators and their descriptions are shown in the following table. Either the alphabetic letters or special character symbols can be used at any time.

Letters	Symbols	Description
EQ	=	is equal to
NE	*=	is not equal to
GT	>	is greater than
LT	<	is less than
EG/GE	=>	is equal to or greater than
EL/LE	=<	is equal to or less than
NG	*>	is not greater than
NL	*<	is not less than

value

Enter a numeric value or a character literal.

A numeric value can include a leading minus sign or an embedded decimal point when appropriate.

A character literal can be enclosed in single quote marks (' '). It must be enclosed in quote marks if it includes blank spaces or special characters. For example:

```
SELECT ELAPS TIME GT 1800 OR APPL CODE(1,3) EQ 'PDN'
```

keyword value

BLANK, BLANKS, SPACE, SPACES, and LOVALUE are special system reserved words. CA JARS Wizard checks for a data field containing nothing but blank spaces or binary zeros. For example:

```
SELECT DYNAM/USERID EQ BLANK
```

This selects records in which the DYNAM USERID field contains a blank space. In this case, the field is eight characters, but it may be any length.

AND and OR

A SELECT statement can have 60 conditional tests connected by AND or OR. If two conditional tests are connected by AND, both must be satisfied for the record to be selected. If they are connected by OR, the record is selected if either is satisfied. Parentheses can be used to group related conditional tests.

Example:

```
SELECT ELAPS TIME GT 1800 OR ACTIVE TIME GT 900 AND
APPL CODE EQ 'ABC123'
SELECT ELAPS TIME GT 1800 OR (ACTIVE TIME GT 900 AND
APPL CODE EQ 'ABC123')
SELECT (ELAPS TIME GT 1800 OR ACTIVE TIME GT 900) AND
APPL CODE EQ 'ABC123'
```

The first and second statements achieve identical results. CA JARS Wizard selects two types of records:

- all records with elapsed time greater than 1,800 seconds
- all records for application code ABC123 that have active time greater than 900 seconds

The third statement selects a smaller group of records: the application code must be ABC123, and either one of the two time tests must be satisfied (it doesn't matter which one).

ALL

indicates that all primary input records are to be selected for processing, irrespective of record type.

MATCHED n

specifies that primary input records with a match on the secondary field are to be selected.

n is a value from 1 through 3 used to refer to the matching function against the first secondary file, the second secondary file, or the third secondary file. If n is omitted, all nonglossary files are considered in the selection process

MISSING n

The primary input records without a match on the secondary file are to be selected. n is a value from 1 through 3, as described for MATCHED n.

SAME field-name

specifies that if two or more consecutive records contain identical values on the specified field name, the second and subsequent records containing the identical value are selected.

NEW field-name

specifies that the first record in a control group defined by the specified field name is to be selected.

Considerations

- A run can include multiple SELECT, SELECT(S), and SELECT(L) statements.
- Statements at different levels are processed separately. For example, SELECT(S) statements are handled independently of SELECT and SELECT(L) statements in the run.
- SELECT statements at the same level are assumed to be connected by AND. OR is only assumed when the same field name appears more than once.

Example:

```
SELECT WEEKDAY EQ SUNDAY AND ELAPS TIME GT 300
SELECT WEEKDAY EQ SATURDAY
```

These two statements have the same effect as the following single statement:

```
SELECT (WEEKDAY EQ SUNDAY AND ELAPS TIME GT 300) OR
WEEKDAY EQ SATURDAY
```

Only one SELECT SAME, SELECT NEW, COMPUTE IF SAME, or COMPUTE IF NEW may be coded for a given field.

SORT Statement

Introduction

The SORT statement indicates how the primary input records are to be sorted for processing.

Format

```
SORT field-name1[(x,y)][(D)]
[, field-name2[(x,y)][(D)],..., field×name9[(x,y,)][(D)]]
[WORK=n] [OPTION='sort-options']
```

Parameters

field-name

All data field names must be for fields in the input file or for work fields that are defined and computed before the SORT is executed. A name can be qualified, as in JOB NAME(1,3). Enter up to nine data field names in descending order of importance.

(D)

Sorting is normally performed in ascending order, from A-Z and 0-9. A field is sorted in descending order if its name has a (D) suffix.

WORK = n

is used in a SORT statement when SYS001 does not provide enough work space. Additional devices may be assigned on SYS002, SYS003, etc. n specifies the total number of devices.

Example:

```
SORT JOB NAME WORK = 2
```

This parameter instructs CA JARS Wizard to use SYS001 and SYS002 for work space.

Note: This parameter applies to VSE only.

OPTION =

allows you to pass optional parameters to the installation's SORT system.The parameters should be enclosed with apostrophes and follow the conventions of the SORT system.

Considerations

The SORT statement is optional. If omitted, CA JARS Wizard processes records in the same order that they are read from the input file.

If the OPTION = parameter is specified, it must be separated from the last field name in the sort list by at least two blanks.

A SORT statement is required in any run that has a BREAK statement if the file is not in sequence. The data field names can be omitted if they are the same as those in the BREAK statement.

Example:

```
SORT
BREAK JOB NAME, STEP NAME
```

The SORT statement in this example is equivalent to one that reads SORT JOB NAME, STEP NAME.

Sample Usage

```
SORT DATE, JOB NAME
```

This sample statement sorts records into order by job name within the run date. The report contains all records for May 1 followed by those for May 2, etc. For any given date, records are in alphanumeric order by job name.

```
SORT DATE(D), JOB NAME
```

This example is the same as above, except that the report lists records for May 2 before those for May 1, etc. Jobs are still listed in ascending order, from A-Z, 0-9 as before.

TITLE Statement

Introduction

The TITLE statement provides a line of text to be printed over two or more adjacent data columns. The effect is to link a group of fields containing related data.

Format

```
TITLE n1-n2=text
```

Parameters

n1

is the starting column.

n2

is the ending column.

text

can be a character literal. CA JARS Wizard pads it on both sides with dashes, if necessary, to stretch between the designated columns.

Sample Usage

```
LIST DATE, DIV/CODE, CHARGE/CODE, APPL/CODE, ELAPS/TIME
TITLE 2-4=USER ID
```

The top of each report page appears as follows, assuming there are no HEADING lines:

```
              *---------------USER ID--------------*
DIV           CHARGE         APPL         ELAPS
   DATE       CODE          CODE           CODE         TIME
```

TITLE(V) Statement

The TITLE(V) statement provides a title for a column of vertical totals.

Format

```
                [                      ]
TITLE(V) n1=text[, n2=text,..., n6=text]
                [                      ]
```

Parameters

n

The column number can be a value from 1 through 6.

text

Specify the column title.

XTAB Statement

Introduction

An XTAB statement causes CA JARS Wizard to generate a cross-tabulation report.

Format

```
XTAB field-a[(F)]=field-a-value1[, field-a-value2,..., field-a-value13]
* field-b[(F)]=field-b-value1[, field-b-value2,..., field-b-value39]
```

A maximum of ten XTAB control statements can be specified in a run.

Parameters

field-a

This is the data item name for the x-axis of cross tabulation. It can only be in character format, excluding dates, and a maximum length of eight characters.

field-b

This is the data item name for the y-axis of cross tabulation. It can only be in character format, excluding dates, and a maximum length of 8 characters.

field-a-value

These are data item values for field-a. Up to 13 can be specified for the x-axis field values (field-a-values). If none are specified, the first 13 values encountered in the file are used.

field-b-value

These are data item values for field-b. A maximum of 39 values can be specified for the y-axis field values (field-b-values). If none are specified, the first 39 values encountered in the file are used.

(F)

Field values are ordered in decreasing total order, instead of alphabetic or user code order.

Chapter 4: Functional Characteristics

This chapter describes the various functions and features of CA JARS Wizard, with emphasis on when functions are used. Topics include:

- Use of the Glossary
- Input Files
- File Matching
- Selecting and Bypassing Records
- Sorting
- Control Breaks and Group Totals
- Work Fields
- Calculations
- Automatic Processing Functions
- Report Functions
- Output Files
- Hexadecimal Fields and Bit Testing

Note: Only one report can be produced by a single CA JARS Wizard run, unless the EXCEPTION, LIST(both), PLOT, or RANK statement is used.

This section contains the following topics:

21st Century Support

The overall design of 21st century support in CA JARS Wizard is to treat years with values of 00 through 59 as being in the 21st century; for example, 2000-2059, and years with values of 60 through 99 as being in the 20th century; for example, 1960-1999. In order for a field to be processed and displayed in this manner, the field must be defined with an attribute identifying it as a date field. Additionally, in order for CA JARS Wizard to take the century into account when sorting, comparing, or performing selection/rejection on a field, it must be defined with the YYDDD, YYMMDD, or YY/MM/DD attribute.The following text discusses in detail each of the areas of date processing affected by 21st century support.

Displaying Date Fields

When a field is printed that is defined as a date, the year portion of the date is checked to see whether or not the year is less than 60. If the year is less than 60, CA JARS Wizard displays the year as being in the 21st century; for example, a year of "02" is displayed as 2002. If the year is greater than or equal to 60, CA JARS Wizard treats it as being in the 20th century; thus, a year of "98" is displayed as 1998.

Sorting on Date Fields

When a date field that is in YYDDD, YYMMDD, or YY/MM/DD format is specified in a SORT statement, the century is taken into account in setting up the SORT field.If the year is 59 or less, it is treated as being in the 21st century; if the year is 60 or higher, it is treated as being in the 20th century.Thus, when performing a sort in ascending sequence, CA JARS Wizard sorts a YYDDD field containing "99364" (December 30, 1999) ahead of a field containing "00003" (January 3, 2000).

Selection, Rejection, and Compares Involving Date Fields

When performing selection or rejection based on a date field defined as in YYDDD, YYMMDD, or YY/MM/DD format, CA JARS Wizard takes the century into account. If the year portion of the date contains a value of 59 or less, it is treated as being in the 21st century; if the year is 60 or more, the date is treated as being in the 20th century.CA JARS Wizard makes the same assumption when comparing a date in YYDDD, YYMMDD, or YY/MM/DD format with a constant.Thus, for selection, rejection, or comparison purposes, a YYMMDD date containing "000103" (January 3, 2000) compares high against a constant of "991230" (December 30, 1999).

Calculations Involving Date Fields

CA JARS Wizard takes century into account when resolving the LODATE and HIDATE keywords. As with other operations involving dates, years of 59 or less are treated as being in the 21st century and years of 60 or more as being in the 20th century.Thus, in a file containing dates ranging from December 27, 1999 through January 7, 2000, LODATE is resolved as 12/27/99 and HIDATE as 01/07/00.

Likewise, CA JARS Wizard takes century into account in resolving the WEEKDAY and DAY OF WEEK keywords.Again, if the year is 59 or less, it is assumed to be in the 21st century. If the year is 60 or higher, it is assumed to be in the 20th century.So a MM/DD/YY date containing "01/01/97" generates a WEEKDAY value of WEDNESDAY, while a date containing "01/01/00" generates a WEEKDAY value of SATURDAY.

Date Manipulation

Where user-coded CA JARS Wizard control statements manipulate dates or portions of dates, these statements may need to be modified in order to process 21st century dates correctly.For example, if a COMPUTE(P) statement moves the year portion of an MM/DD/YY date to a field specified in a SORT statement,CA JARS Wizard treats that sort field like any other character field and makes no adjustment for century; thus, a value of 00 sorts ahead of a value of 99.

The same holds true for selection/rejection on a field not defined as a date. If a SELECT or REJECT statement specifies a year field (not an entire date field in YYDDD, YYMMDD, or YY/MM/DD format), CA JARS Wizard does not take the century into account and thus treats a year of 00 as less than a value of 98. The control statements have to be modified in order for the century to be taken into account.

Thus, if a date or portion of a date is moved to a field not defined with a date attribute, the target field is treated as an ordinary number or character string. It makes no adjustment for century when processing the field.

Use of the Glossary

Glossary Files

Every file to be processed by CA JARS Wizard must be defined. Each definition consists of a file and a glossary definition:

File Definition

A predefined entry exists in CA JARS Wizard, specifying the following information for all supported primary files:

- FILE NAME - The name of the glossary that contains further information about the records in the file.
- RECORD LENGTH - The maximum record length of a record in the file.
- BLOCKING FACTOR - The number of logical records that make up a physical block.
- RECORD FORMAT - The format of the records in the file, as follows:

  F

  for fixed unblocked

  FB

  for fixed blocked

  V

  for variable unblocked

  VB

  for variable blocked

  VBS

  for variable blocked spanned

  U

  for undefined

Glossary Definition

A subroutine, consisting of the glossary definition for the file, is supplied by CA to enable CA JARS Wizard to access a primary file.

The glossary definition consists of the following entries:

- FIELD NAME - The name of the field. Each field in the record is assigned a field name. References to the field can be made using this name in any CA JARS Wizard statement.

  The field name is also used as the default column heading when printing the field.

- START POSITION - The starting position of the field within the record.
- LENGTH - The length of the field.

- FORMAT - The format of the field. CA JARS Wizard allows the following formats:

  C

  character

  N

  numeric, zoned decimal

  B

  binary

  P

  packed decimal

  H

  hexadecimal

- SCALING - The number of decimal positions in the field, if any. The format must be N, P, or B.
- PRINT CHARACTERS - The number of print positions that should be allocated for the field. If omitted, the default assumed is the number of print positions required by the length specification and format of the field. For example, a field defined as 4 bytes binary is automatically allocated at least 10 print positions for the maximum decimal value it can contain. Thus, if you expect the field to contain less than the maximum capacity of the field, the number of print characters should be specified in the LIST statement.
- Time or Date Attributes

  Date Attributes - The internal format of the date field.

  Gregorian Format

  YY/MM/DD
  MM/DD/YY
  DD/MM/YY
  YYMMDD
  MMDDYY
  DDMMYY

  Julian Format

  YYDDD

  Time Attributes - The format for display of the time field.

  HM

  Displayed as HH.MM

HS

Displayed as HH.MM.SS

MS

Displayed as MM.SS

HH

Displayed as hundredths of hours

KH

Displayed as HH.MM.SS.SSS

KM

Displayed as MM.SS.SSS

KS

Displayed as SS.SSS

- Other Attributes - Additional specifications can be defined in the glossary. These are discussed in the "Appendix B" chapter.

  You can display the file and glossary definitions included in each of the products.

  See the "Appendix D (see page 175)" chapter for a description of the product glossaries. The "Sample Usage (see page 19)" chapter provides information on creating a listing of a CA JARS Wizard glossary.

  Note: Refer to the *CA JARS Installation Guide* for a listing of the file and glossary names available by CA JARS Wizard.

Record Type Qualification

Certain product glossaries may describe records of only one type. These have fixed field references. Field names in such glossaries are unqualified, as each field is defined in every record of the file.

Glossaries that describe multiple record type files contain field names with record type prefixes. In such files, each record consists of two portions:

- a common portion that contains constant data, including a record type descriptor, and
- a variable portion whose contents depend solely upon the record type descriptor in the fixed portion.

All data field names in the common portion are unqualified by record type as they are defined in every record of the file. However, all data field names in the variable portion must be prefixed by the record type in order to indicate in which record types the field exists.

Example:

E.CPU TIME references the field CPU TIME in the E-type record
L.CPU TIME references the field CPU TIME in the L-type record

Note: The same field can exist in more than one type of record. CA JARS Wizard automatically separates these references as long as they are correctly prefixed with the record type.

Because the O prefix is used to denote output glossary fields, (refer to the description of the OUTPUT (see page 100) statement in Chapter 3), no glossary contains O-type records.

Global Record Type Qualification

Whenever record type qualification is required, you can elect to reference every occurrence of a particular data field, regardless of the record types in which it is defined. This global capability is achieved by prefixing the field name with an asterisk (*).

Example:

E.CPU TIME references the field CPU TIME in the E-type record.
*.CPU TIME references the field CPU TIME in all record types in which it is defined.

Output Record Qualification

Whenever an output glossary file is created, all fields in the output record are referenced by the prefix O.

Example:

O.E.CPU TIME if the field requires record type qualification
O.JOB NAME if the field does not require record type qualification

Input Files

CA JARS Wizard processes one file at a time, and can access a total of four input files. When performing matching functions, however, up to three secondary files can be specified.

Glossary Files

If a glossary file is to be processed, then this is specified by an INPUT glossary-name statement. The reference to the glossary provides you with the facility to refer to predefined fields in the file by means of field names. All of the attributes defined in the glossary for the fields are used for processing defaults.

Note: For more information, see the *CA JARS Installation Guide* for details on file glossaries.

Nonglossary files

Nonglossary files are defined by the INPUT statement with the operand FILE, CARDS, or member-name. When no glossary file is included in the same CA JARS Wizard run, a maximum of four such files can be defined. When you are performing file matching functions, up to three nonglossary files can be defined in addition to the glossary file.

Reference to fields in nonglossary files is made through fCCa-b notation, where a is the starting position and b is the ending position of the field. f is the file sequence number and has a value from 1 to 4. The value of f corresponds directly with the physical sequence of the input files as follows:

with two glossary files:

first glossary file	f = 1 is assumed by CA JARS Wizard
first nonglossary file	2CCa-b
second glossary file	f = 3 is assumed by CA JARS Wizard
second nonglossary file	4CCa-b

without a glossary file:

first nonglossary file	1CCa-b
second nonglossary file	2CCa-b
third nonglossary file	3CCa-b
fourth nonglossary file	4CCa-b

If only one nonglossary file is present, f may be omitted.

Note: The first physical input file, whether glossary or nonglossary, is considered to be the primary input file (f = 1). All computations, selections, and so forth, are performed against this file only. All other files, if any, are considered to be secondary files.

Example:

```
INPUT FILE RECFM=F R=105
COMPUTE(P) DUP IND(C)(5) = 1CC1-3,
        REC TYPE(C) (2) = 1CC6-7,
        ITEM/NAME(C) (16) = 1CC8-23,
        POSITION(N) (3) = 1CC26-28,
        LENGTH(C) (3) = 1CC31-33,
        FORMAT(C) (6) = 1CC36-41,
        SUFFIXES(C) (12) = 1CC44-56,
        ITEM/DESCRPT(C) (42) = 1CC63-105,
        FILLER1(C) (2) = BLANKS,
        FILLER2(C) (6) = BLANKS
SORT REC TYPE, POSITION
BREAK REC TYPE
OUTPUT FILE ITEMS = DUP IND, FILLER1,
             REC TYPE,
             ITEM/NAME, FILLER1,
             POSITION, FILLER1,
             LENGTH, FILLER1,
             FORMAT, FILLER1,
             SUFFIXES, FILLER2,
             ITEM/DESCRPT
```

Note: For more information, see the discussion of the INPUT (see page 91) statement in the "Control Statements" chapter.

File Matching

To perform file matching, CA JARS Wizard processes data from a primary input file and up to three secondary files in a run. A secondary file can be either of the following:

- A table that is read into storage and searched as each primary input file record is processed.
- A data file, properly sorted, that is read record by record, and matched with records from the primary input file, based upon a common data field.

The primary input file should be defined by a glossary. Secondary files do not have to be defined by a glossary since items can be referenced in general or by DEFINE statements in the run.

CA JARS Wizard has an fCCa-b notation to refer to data fields in secondary files. 2CCa-b defines positions in the first secondary file. 3CCa-b refers to the second, and 4CCa-b to the third. For example, 2CC20-25 refers to positions 20 to 25 in the first secondary file, and 3CC20-25 refers to the same positions in the second secondary file.

The keywords MATCHED and MISSING can be used in SELECT and COMPUTE statements to determine whether a match has occurred between the primary input file and the secondary files. Processing can then be directed accordingly, based upon this determination.

Example:

```
INPUT PRIMARY-INPUT-FILE
SELECT REC/TYPE EQ 'E'
INPUT FILE KEY=2CC1-4 MATCH=APPL CODE TABLE=2CC5-20
INPUT FILE KEY=3CC1-4 MATCH=APPL CODE
SELECT MATCHED1 AND MATCHED2 AND DIV CODE EQ 'T' AND MONTH EQ '05'
COMPUTE APPL NAME(K)(C)(16) EQ 2CC5-20,
        PROGRAMMER TIME(N)(4) EQ 3CC48-51
SORT APPL CODE
BREAK APPL CODE
LIST(SUM) APPL NAME, APPL CODE, PROGRAMMER TIME, ACTIVE TIME
```

Use this example when introducing secondary input to the run. When INPUT is used it sets up the "key" pointing to the secondary input file and also indicates what it should "MATCH" in the glossary. COMPUTE if matched will match based on what criteria you give it.

This example assumes the following:

- The first secondary file has an application code in positions 1-4 and a system development application name in positions 5-20.
- The second secondary file contains an application code in positions 1-4. Positions 48-51 contain the total number of programmer hours for the project for the month of May.
- The first secondary file doesn't require any sorting since the TABLE parameter initiates binary lookup procedure.
- The second secondary file, however, must be sorted in key sequence, (positions 1-4 of the record), prior to the CA JARS Wizard run.

The following processing actions are made in this example (the exact sequence of operations varies from run to run, according to the specific requirements for the run):

- The primary input file is sorted by application code (SORT statement). This sort is necessary because of the record-for-record matching with the secondary input file.
- The entire first secondary file is read into storage and sorted. The TABLE parameter in this file's INPUT statement tells CA JARS Wizard that it is a table lookup file. A binary search is used to locate matches.

- A record is read from the primary input file and evaluated based upon the selection criteria. The record is selected if the DIV CODE is a T, the month portion of the date field is 05, and CA JARS Wizard is able to find matching records in both of the secondary files.
- CA JARS Wizard can now reference two data fields of records where a match occurred with the two secondary files. They are the application name field in positions 5-20 of the first secondary file and the programmer time field in positions 48-51 of the second secondary file. The COMPUTE statement moves to these fields.
- Normal processing is performed for the primary input record.
- The primary file could contain multiple records for the same application code. Consequently, the application name requires a (K) suffix, because character fields that are not break items are not printed on total lines (this is the default). The programmer time work field is defined as numeric data so that it is accumulated as a volume and displayed in the summary report print line.

The product of this example would be a summary report with the month's totals for each application code. The line would contain the application name, the application code, the total programmer hours from the tape file, and total active time from all records for the application in the file.

Note: For more information about multiple file processing, see the INPUT (see page 91), SELECT (see page 110), and COMPUTE (see page 71) statement discussions in the "Control Statements" chapter.

Selecting and Bypassing Records

CA JARS Wizard can process all records from an input file. However, the most meaningful reports are produced if they are restricted to records that meet special conditions.

For example, it may be necessary to list data for all job steps that run longer than five minutes. The requirement is expressed in a SELECT statement, as follows:

```
SELECT ACTIVE TIME GT 300
```

This sample statement causes this product to examine the data in the ACTIVE TIME field of each record in the input file. If the value is greater than 300 seconds, it continues processing the record. If not, it bypasses the record.

None of these actions has any effect on the input file itself. The input file is not altered by use in a CA JARS Wizard run.

The result of a SELECT operation is that the run processes only those records that meet the conditions specified by the SELECT statement.

Selection can be based on any number of conditions.

Example:

```
SELECT ACTIVE TIME GT 600 AND LINES PRINTED LT 1000 AND
CHARGE CODE EQ 'B5' OR CHARGE CODE EQ 'B6'
```

This sample statement selects only those records in which the active run time is greater than 600 seconds or ten minutes, the number of print lines is less than 1,000, and the CHARGE CODE is either B5 or B6. All other records in the file are ignored.

The use of selection criteria is particularly important when processing a large file. A complete listing of such a file can produce a large volume of output. The SELECT statement reduces the volume by restricting the scope of the report.

CA JARS Wizard also has a REJECT statement. It can be used in place of SELECT when it is easier to specify which records to bypass. For example, the following two statements have identical results:

```
SELECT ACTIVE TIME GT 300
REJECT ACTIVE TIME LE 300
```

Selection and rejection may be performed at various times in the processing cycle. A suffix code designates the time, for example:

SELECT(P)/REJECT(P)

Record selection takes place before the input file is sorted and before any preSORT computations.

Note: The run MUST contain a SORT statement.

SELECT/REJECT

Record selection takes place before the input file is sorted, (assuming the run contains a SORT statement).

SELECT(S)/REJECT(S)

Selection takes place after the SORT.

SELECT(L)/REJECT(L)

Selection criteria in these statements applies solely to the report listing. Records can be selected for processing and bypassed for printing. (This applies to detail lines only.)

Sorting

Input records can be sorted into any order for processing. CA JARS Wizard uses from one to nine data fields as sort keys. For example:

```
SORT DIV CODE, ACTIVE TIME
```

This sample statement sorts records in alphanumeric order, from (A-Z and 0-9), by division code. Within each division, records are organized in numerically ascending order by active time.

The (D) suffix after a data field name provides a descending sort.

Example:

```
SORT DIV CODE, ACTIVE TIME(D)
```

Records are sorted in alphanumeric order by division code, as before. In this case, however, records within each division are sorted numerically in descending order of active time. Those with the largest active time values appear first.

A SORT statement is required in any run that includes a BREAK statement if the file is not in the proper sequence. Data field names must be in the same order that they appear in the BREAK statement. The SORT statement, however, can include additional field names that are not needed in the BREAK statement.

A data field name may be qualified.

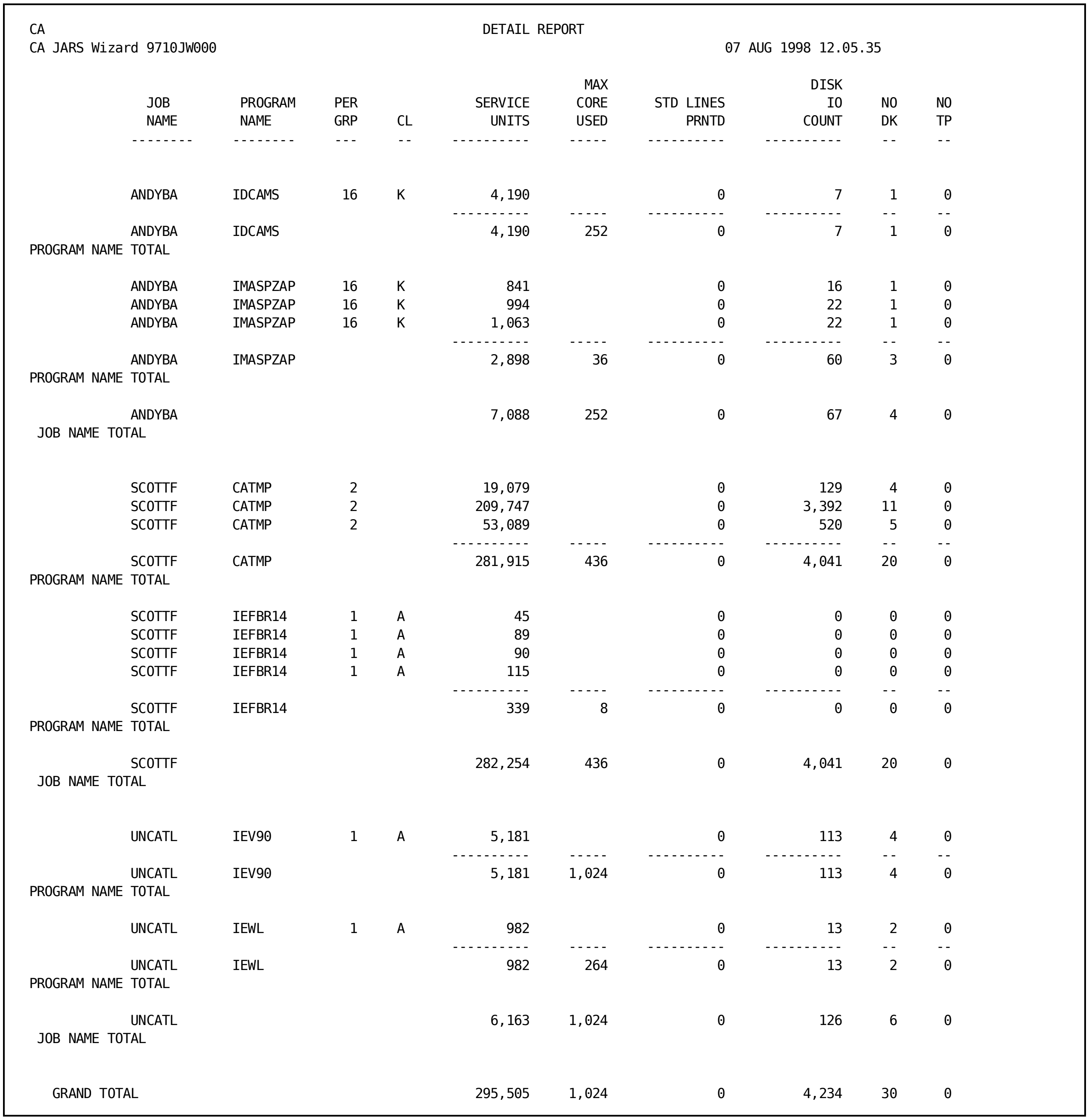
Example:

APPL CODE (1,3) would direct consideration of only the first, second, and third characters in the application code data field for sorting.

Control Breaks and Group Totals

CA JARS Wizard can provide for up to eight control breaks and group totals. For example, an installation may want a report displaying CICS/VS activity by run date, transaction ID, and operator ID. It must also show transaction use count, CPU time, dispatch time, terminal input count, total file IOs, and average response time. The control statements and the report created by them are given below.

```
INPUT primary-input-file
SELECT PRO ID EQ ('S' OR 'T') AND STEP JOB IND EQ (' 1' OR '11')
SORT JOB/NAME, PROGRAM/NAME
BREAK JOB/NAME, PROGRAM/NAME
LIST JOB/NAME, PROGRAM/NAME, PER/GRP, CL, SERVICE/UNITS,
     CORE/USED(MX), STD LINES/PRNTD, DISK/IO/COUNT, NO/DK, NO/TP
```



```
CA                                                        DETAIL REPORT
CA JARS Wizard 9710JW000                                                                 07 AUG 1998 12.05.35

                                                                       MAX                          DISK
               JOB         PROGRAM     PER               SERVICE      CORE      STD LINES             IO     NO     NO
               NAME        NAME        GRP     CL          UNITS      USED          PRNTD          COUNT     DK     TP
             --------     --------     ---     --     ----------     -----     ----------     ----------     --     --


             ANDYBA       IDCAMS        16     K           4,190                        0              7      1      0
                                                      ----------     -----     ----------     ----------     --     --
             ANDYBA       IDCAMS                           4,190       252              0              7      1      0
PROGRAM NAME TOTAL

             ANDYBA       IMASPZAP      16     K             841                        0             16      1      0
             ANDYBA       IMASPZAP      16     K             994                        0             22      1      0
             ANDYBA       IMASPZAP      16     K           1,063                        0             22      1      0
                                                      ----------     -----     ----------     ----------     --     --
             ANDYBA       IMASPZAP                         2,898        36              0             60      3      0
PROGRAM NAME TOTAL

             ANDYBA                                        7,088       252              0             67      4      0
 JOB NAME TOTAL


             SCOTTF       CATMP          2                19,079                        0            129      4      0
             SCOTTF       CATMP          2               209,747                        0          3,392     11      0
             SCOTTF       CATMP          2                53,089                        0            520      5      0
                                                      ----------     -----     ----------     ----------     --     --
             SCOTTF       CATMP                          281,915       436              0          4,041     20      0
PROGRAM NAME TOTAL

             SCOTTF       IEFBR14        1     A              45                        0              0      0      0
             SCOTTF       IEFBR14        1     A              89                        0              0      0      0
             SCOTTF       IEFBR14        1     A              90                        0              0      0      0
             SCOTTF       IEFBR14        1     A             115                        0              0      0      0
                                                      ----------     -----     ----------     ----------     --     --
             SCOTTF       IEFBR14                            339         8              0              0      0      0
PROGRAM NAME TOTAL

             SCOTTF                                      282,254       436              0          4,041     20      0
 JOB NAME TOTAL


             UNCATL       IEV90          1     A           5,181                        0            113      4      0
                                                      ----------     -----     ----------     ----------     --     --
             UNCATL       IEV90                            5,181     1,024              0            113      4      0
PROGRAM NAME TOTAL

             UNCATL       IEWL           1     A             982                        0             13      2      0
                                                      ----------     -----     ----------     ----------     --     --
             UNCATL       IEWL                               982       264              0             13      2      0
PROGRAM NAME TOTAL

             UNCATL                                        6,163     1,024              0            126      6      0
 JOB NAME TOTAL


   GRAND TOTAL                                           295,505     1,024              0          4,234     30      0
```

Highlights of the sample report are as follows:

- Job statistics are displayed for a variety of jobs followed immediately by their individual job name totals. Records included within each of these jobs are split out by programmer name with totals displayed for each program.
- The control break that occurs on changes in PROGRAM/NAME is referred to as the lowest, or first level control break. The control break for JOB/NAME is referred to as the highest (or, in this case, the second) level control break.

- CA JARS Wizard automatically skips lines after each set of control group totals before printing the first line of the next group. If you use three or four BREAK fields in a report, it skips to a new page after breaks for the two highest levels.You can override the automatic skipping.
- CA JARS Wizard determines what fields to print in a totals line, based upon the format characteristics of each field. Suffix codes are provided that can be used to override normal totalling procedures when necessary. These codes are covered in the description of the LIST statement and in the "Appendix B (see page 155)" chapter.

The description of the BREAK statement identifies other facilities for the totals line and for page formatting.

Vertical Totals

Additional totals can be requested for control break groups by using suffix codes in the LIST statement. A (T), (Tn), or (TG) suffix after the field name causes CA JARS Wizard to compute totals for fields that are not included in the print line, and to list them after the totals line for a break group, as follows:

(T)

A total for the field is printed after every control break.

(Tn)

Specify n as a control break level from 1 through 8. The total for the field is printed only at the specified and higher control break level. One is the lowest level, and eight is the highest possible level.

(TG)

A total for the field is printed only with grand totals at the end of the report.

Example:

```
INPUT primary-input-file
SELECT REC TYPE EQ 'E'
SORT CHARGE CODE, JOB NAME
BREAK CHARGE CODE, JOB NAME
LIST JOB NAME, ELAPS TIME, CPU TIME(T), NO OF JOBS(T2)
```

The following listing might be produced by this example:

```
®                              JOB NAME          ELAPS TIME       ®
®                                                                 ®
®                              ABC123                12.34        ®
®                              ABC123                13.24        ®
®                                                __________       ®
®   JOB NAME TOTAL                                   25.58        ®
®                                                                 ®
®     CPU TOTAL  1.23                                             ®
®                                                                 ®
®   CHARGE CODE TOTAL                                25.58        ®
®                                                                 ®
®     CPU TOTAL  2.30                                             ®
®                                                                 ®
®     NO OF JOBS    2                                             ®
®                                                                 ®
```

In this example, the CPU TIME field has a (T) suffix, so its total appears in every control break. It also prints with grand totals.The NO OF JOBS total appears only with the CHARGE CODE totals at control break level 2.

Work Fields

New data fields can be established in any run. These fields can be used temporarily, printed in a report, or included in an output file. Their values can be used for selection and sorting or for other computations, in the same way as data values from the input file.

Work fields can be defined in the DEFINE, DECODE, or COMPUTE statements.

During the definition of the work field, format characteristics are assigned for the work field. These may be defined explicitly by you or the system default characteristics can be used. The work field retains the format for the rest of the run.

Format characteristics that are associated with a work field name are:

Data Format

character, packed, numeric, binary, or hexadecimal

Length

number of bytes

Scaling

number of decimal positions

Work Fields and the DEFINE Statement

The DEFINE statement can be used to create work fields and initialize them with a numeric or character value.

```
FORMAT:   DEFINE fieldname EQ nnn
DEFINE fieldname EQ 'character string'
```

Unless otherwise indicated, the field is always reset to the initially assigned value whenever a new input record is processed. CA JARS Wizard also performs automatic accumulation for the field throughout the run and displays the work field's total on total lines.

Work fields created with the DEFINE statement cannot be used in preSORT statements, nor in the SORT or BREAK statements.

The following default formats and lengths are used:

Constant	Format	Length
Numeric	Packed	4 bytes
Literal	Character	4 bytes or as required by the literal

If the numeric value assigned is scaled, then the field is given a scaling characteristic.

If you do not like the above defaults, then you can specify explicitly the format, length, and scaling characteristics for the work field. For example:

DEFINE	Format	Length	Scaling
RATE EQ .75	PACKED	4 bytes	2 decimal places
FACTOR EQ 150	PACKED	4 bytes	
NAME EQ 'TOTAL JOBS'	CHARACTER	10 bytes	
CODE(H)(1) EQ 'FF'	HEXADECIMAL	1 byte	

Work Fields and the DEFINE(K) Statement

DEFINE(K) is a special form of the DEFINE statement that does essentially what the DEFINE statement does, except that the value assigned to the work field is not reset when a new record is processed. The value of the field remains unchanged unless you explicitly change it with a COMPUTE statement. CA JARS Wizard does not accumulate totals for work fields created with DEFINE(K).

Example:

```
DEFINE ACTR EQ 5
DEFINE(K) BCTR EQ 5
COMPUTE ACTR EQ ACTR + 1,
        BCTR EQ BCTR + 1
LIST ACTR, BCTR
```

In this example, the ACTR column in the report contains 6 for the first record, 6 for the second record, 6 for the third record, etc.

The BCTR column, however, contains 6 for the first record, 7 for the second record, 8 for the third record. Assuming only three records in the file, at grand total time, ACTR contains 18, BCTR contains 8.

Work Fields and the COMPUTE Statement

Work fields that were not previously defined in a DEFINE or DECODE statement are created and initialized when first encountered in a COMPUTE statement.

The work field is assigned format characteristics based on your explicit specifications, or defaults to PACKED format, 4 bytes in LENGTH, and 0 SCALING. The work field retains its format characteristics throughout the run.

The work field is initialized with zeros or blanks, depending on whether the data format is numeric or character. The value of the work field changes depending on the COMPUTE logic, but is always reset to zero or blanks for each new record, except if given the suffix (K).

Example:

```
COMPUTE CTRA EQ CTRA + 1,
CTRB(B)(.2) EQ CPU TIME,
CTRC EQ JOB NAME,
CTRD(C)(8) EQ STEP NAME
```

CTRA defaults to PACKED, 4 BYTES, 0 SCALING. Its initial value is 0; first record value is 1; second value is 1; third record value is 1, etc.

CTRB is defined as BINARY, 4 BYTES, 2 DECIMAL PLACES. Its initial value is 0; first record value is the first input record's CPU TIME; second record value is the second input record's CPU TIME, etc.

CTRC causes a diagnostic error because JOB NAME is defined in the glossary with a CHARACTER format and CTRC defaults to PACKED format.

CTRD is assigned the format characteristics of CHARACTER, 8 BYTES. Its initial value is blanks; first record value is the first input record's STEP NAME; second record value is the second input record's STEPNAME; third record value is the third input record's STEP NAME, etc.

The use of the suffix (K) after the data item name causes the definition of the work field to proceed in the same manner as described above, except that CA JARS Wizard does not reset the work field. The work field retains its value until explicitly altered by a COMPUTE statement.

Example:

COMPUTE CTRA(K) EQ CTRA + 1

CTRA defaults to be PACKED, 4 BYTES, 0 SCALING. Its initial value is 0; first record value is 1; second record value is 2; third record value is 3, etc.

Work Fields and the DECODE Statement

Work fields that were not previously defined in a DEFINE or COMPUTE statement are created when first encountered in a DECODE statement.

The work field is assigned format characteristics based on your explicit specifications or defaults to PACKED format, 4 bytes in LENGTH, and 0 SCALING. The work field retains its format characteristics throughout the run.

The value of the work field is either set to zero or is blank at the beginning of each record processing cycle, and is assigned values by the DECODE statement.

Calculations

The COMPUTE statement enables processing instructions in a CA JARS Wizard run. The processing instructions may perform the following:

- modify input fields
- generate new work fields
- perform assignment instructions
- perform arithmetic functions
- convert field contents to other formats
- performs selective processing with IF-THEN-ELSE-ALWAYS statements

- perform a branch to another COMPUTE statement
- exit from a series of COMPUTE statements

COMPUTE statements can also be executed at various points of the processing cycle. A suffix code specifies the time, for example:

COMPUTE(P)

Indicates that the statement should be executed before the SORT is performed. Fields generated at this time can be used as selection fields or as SORT fields.

COMPUTE

Indicates that the statement should be executed after the SORT.

COMPUTE(T)

Indicates that the statement should be executed every time there is a control break.

COMPUTE(n)

Where n is a value from 1 through 8, COMPUTE(n) indicates that the statement should be executed during the control level break and at all higher control level breaks.

For example, COMPUTE(4) means perform the statement at the fourth, fifth, sixth, seventh, and eighth control level break.

COMPUTE(G)

Indicates that the statement should be executed at grand total time.

Note: For further information, see the explanation of the COMPUTE (see page 71) statement in the "Control Statements" chapter.

Automatic Processing Functions

Input Numeric Fields

CA JARS Wizard performs automatic processing on the input numeric fields. These functions include:

- Accumulating control totals if there is a BREAK statement, and grand totals.
- Computing for a control break the maximum, minimum, average, percentage, and standard deviation values for a numeric field using one of the suffixes in the following table.

Suffix	Function
(AV)	Determines the average value of the field by dividing the control level accumulator for the field by the number of records processed in the control group.
(MN)	Determines the lowest value encountered for the field in the control group.
(MX)	Determines the highest value encountered for the field in the control group.
(PC)	Determines what percentage of the field's total file value comprises the control group's field total value.
(SD)	Automatically computes the standard deviation of the field based upon the values in the relevant control group.
(TL)	Computes the total value of the field at input selection time and makes this value available for subsequent computations.

Notes

- None of these automatic processing functions work for numeric fields defined with the (NV) suffix.
- When you invoke some of these automatic functions, a SORT statement must be included in the run because the processing of some of these functions involves preSORT computations.

  The (PC) and (TL) suffix codes require preSORT totalling; these preSORT totals may include records that are rejected after the SORT. In addition, computations occurring after the SORT are **not** reflected in the calculation.

Example:

If the following statements are included in a given run:

```
COMPUTE(P) FIELDA = X       Computation occurs preSORT.
COMPUTE FIELDA = Y  Computation occurs postSORT.
LIST FIELDA(PC)
```

The LIST statement displays a percentage for value X only.

- All of these suffix codes and suffix codes (CT), (DF), and (DL) are mutually exclusive. They cannot be specified together in an individual reference for a particular field. They can, however, be specified in different references for the same field.

Example

```
INPUT primary-input-file
SELECT REC TYPE EQ 'E'
BREAK JOB NAME
SORT
LIST(SUM) JOB NAME,
               CPU TIME,
               CPU TIME(MX),
               CPU TIME(MN),
               CPU TIME(AV),
               CPU TIME(PC)
```

Assume that the records in the files are:

Rec#	JOB NAME	CPU TIME
1	A	5 seconds
2	A	3 seconds
3	A	1 second
4	B	3 seconds
5	B	18 seconds
6	B	6 seconds

The report is as follows:

```
MAX       MIN        AVG       PCT
               CPU      CPU       CPU        CPU       CPU
 JOB NAME      TIME     TIME      TIME       TIME      TIME
 --------      ----     ----      ----       ----      ----
 A              9        5         1          3         25
 B              27       18        3          9         75
 GRAND TOTAL    36       18        1          6         100
```

Report Functions

CA JARS Wizard has extensive report printing capabilities. Reports are generally produced using the LIST statement. The basic report format consists of columns of fields being printed from left to right on the report. If the suffix (SUM) is used, only summary lines are printed.

In conjunction with the basic LIST capabilities of CA JARS Wizard, additional report functions can be performed using the following statements:

EXCEPTION

Produces a detailed listing of records that meet special conditions.

RANK

Produces reports ranked on different conditions with one pass of the input records.

CALENDAR

Produces a report in calendar format.

Suffix (Gn)

Prints out the value of the field as a horizontal plot. The PLOT statement, which is optional, can be used by you to define the plot format.

XTAB

Produces a table (cross-tabulation) using user-specified data fields.

CA JARS Wizard performs automatic report formatting and control totals accumulation for numeric fields defined in the LIST statement.

Automatic Report Formatting

- CA JARS Wizard automatically prints headings. It uses the FIELD NAME as the column heading, unless you specify the column headings desired in the LIST statement parameters.
- CA JARS Wizard automatically determines the print positions allocated to a field. This is generally based on the requirements of the field, except if you specify a desired print position length. automatic editing of the fields, providing commas and periods for proper punctuation is also performed. Again, this can be overridden if you specify special suffixes after the field names in the LIST statement parameters.
- Page formatting, (such as spacing, number of print positions, and number of lines per page), is done and can be overridden using PARAMETER statement specifications.

Control Breaks

- CA JARS Wizard performs automatic headings for control breaks and grand totals. The headings can be user-defined using an option specified in the BREAK statement.
- CA JARS Wizard also automatically accumulates totals for numeric fields. These automatic accumulations can be overridden by specifying special suffixes after the field name in the LIST statement.
- On control level breaks, only prints character fields that are defined as BREAK fields are printed. The value of the character field for the control group is printed. All other character fields print as blanks on control total lines.To force printing of character fields on control total lines, the field must be given the suffix (CT), (DF) or (DL), or moved to work areas defined with the (K) suffix. The suffix (K) leaves the data manipulation on these fields to the user, and outputs the current contents of the field at print time.

Output Files

CA JARS Wizard can be used to copy, merge, modify, or summarize data files. It writes records to the new file. Data values can be added, deleted, or modified along the way. Records can be added, deleted, or reblocked.

The following variations are possible:

- Every record in the input file can be copied into a new file, with or without modifications.
- Selected records can be processed and output into a new file.
- An output summary record can be written out at control break time and contain summary information for the control group.
- Selected fields from the input record can be specified to comprise the output record.
- Two files can be merged. CA JARS Wizard reads the file to be updated and copies it into the output file without change. The records from the primary file are processed as specified and added at the end of the output file. Note that the records from the two files are not sorted together in the output file; they are simply appended to each other.
- A new file can be created according to a glossary definition. Fields in the output file glossary are referred to by the O prefix.

Note: For more information on the output functions, see the OUTPUT (see page 100) statement discussion in the "Control Statements" chapter.

Hexadecimal Fields and Bit Testing

CA JARS Wizard can be used to test and set entire hexadecimal fields or, alternatively, to test and set individual bits within a hexadecimal field.

Hexadecimal fields must be defined with the suffix (H), together with another suffix denoting the length of the field. Hexadecimal constants must always consist of an even number of characters.

Example:

```
COMPUTE HEX FIELD(H)(1) EQ 80
```

This statement defines a one-byte hexadecimal field and assigns it the hexadecimal value of 80.

Bit testing can be performed only on fields defined as hexadecimal. Individual bits are identified by the precision suffix, (.n) where n is the bit number relative to 1. Allowable values of n are 1 to 8 inclusive, and bit one is the leftmost bit of the field.

Bit references can test for a particular bit to be on or off (=ON/OFF) or have a value of zero or one (=0/1).

Example:

```
COMPUTE IF HEX FIELD(.3) = ON THEN ----
COMPUTE IF HEX FIELD(.3) = 1 THEN ----
```

The above statements are identical in operation and both test the setting of bit number three in the hexadecimal field, HEX FIELD.

Note: A test of a hexadecimal field is a character-type comparison, that is, the entire field is compared with a single value. Individual bit testing, however, tests individual bits, ignoring the hexadecimal value of the field and all unreferenced bits.

CA JARS Wizard Exits

The CA JARS Wizard exit WIZSETT allows multiple transmittals of NT text to be produced whenever a debit amount exceeds 9999.99.

In the debit transmittal for CA JARS, the debit amount field is six bytes in size with two decimals.This is generally sufficient to meet the needs for most applications.

There are some cases where a single charge for one amount exceeds $9999.99, and when produced by CA JARS Wizard it becomes truncated.For example, $12,345.67 becomes $2345.67, or 234567 in the statement field.

There is now an exit designed to work in conjunction with a few CA JARS Wizard coding requirements that will produce transmittals until the entire calculated charge amount is produced.For example, a calculated charge of $12, 345.67 will produce one transmittal for $9999.99 and a second for $2,345.68.If you list the same field as in the output record, the report will show only the calculated amount of $12,345.67.

To use the WIZSETT exits you must:

1. Add the following call immediately before the input statement

   CALL WIZSETT

2. Define the debit amount field in the output record as a six byte packed field.

To use the WIZSETT exits with CA JARS interfaces you must:

1. Add the following call immediately before the input statement

   CALL WIZSETT

2. Change the debit amount field for the output record. For example, ODEBITAMT must be changed from ODEBITAMT (N)(6) to OBEDITAMT(P)(6)(.2). In the example that follows, ODEBITAMT is on the output item list immediately after OPROGRAMMER.

The following is an example of modified CA JARS Wizard code for use in the DB2 Interface.

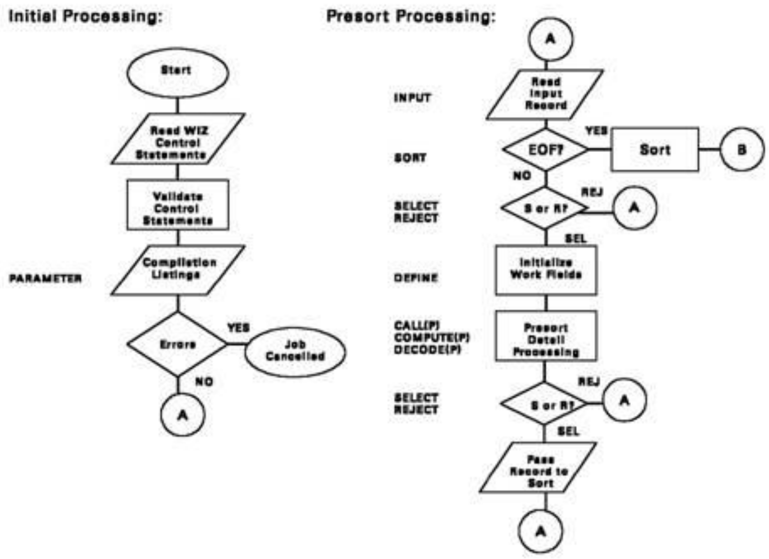
```
PARAMETER NOPC XREF
CALL    WIZSETT
INPUT CAWGJRD2
DEFINE OFILL1(1)(C) = ' '
DEFINE ODEBIT(8)(C0 = 'DEBIT  '
DEFINE OCPUID(1)(C) = ' '
DEFINE OUSERID(8)(C) = ' '
DEFINE ODATE(6)(C) = ' '
DEFINE OJOB(8)(C) = 'DB2-PROC'
DEFINE OTIME(6)(C) = ' '
DEFINE OPROGRAMMER(20)(C) = ' '
DEFINE ODEBITAMT(6)(P)(.2) = 0
*
DEFINE IOREQT(6)(P) = 0
*
INPUT FILE KEY = 2CC1-8 MATCH = USER/ID TABLE=2CC10-25 LIST
COMPUTE(P) IF MATCHED THEN OACCT(C)(16) EQ 2CC10-25 ELSE
OACCT EQ USER/ID
*
COMPUTE OUSERID EQ USER/ID
COMPUTE IOREQT = (READ/SYNC/BP0 + READ/PREF/BP0 +
                  READ/SYNC/BP1 + READ/PREF/BP1 +
                  READ/SYNC/BP2 + READ/PREF/BP2 +
                  READ/SYNC/BP32 + READ/PREF/BP32
*
COMPUTE ODEBITAMT = (CPU/TIME * .19283) + (IOREQT * .0123)
*
SORT OACCT
BREAK OACCT
HEADING1C TOTAL CHARGES BY ACCOUNT
LIST(SUM) OACCT(DL) 'ACCOUNT',
          ODEBITAMT($) 'TOTAL CHARGE',
          CPU/TIME 'CPU TIME',
          IOREQT 'READ/REQUEST',
          COUNT 'DB2 ACCOUNT/RECORDS'
*
OUTPUT(SUM) FILE,RECFM = FB,R = 80,B = 800,
   ITEMS = OFILL1,ODEBIT,OCPUID,OUSERID,OACCT,ODATE,OJOB,
   OTIME,OPROGRAMMER,ODEBITAMT
```

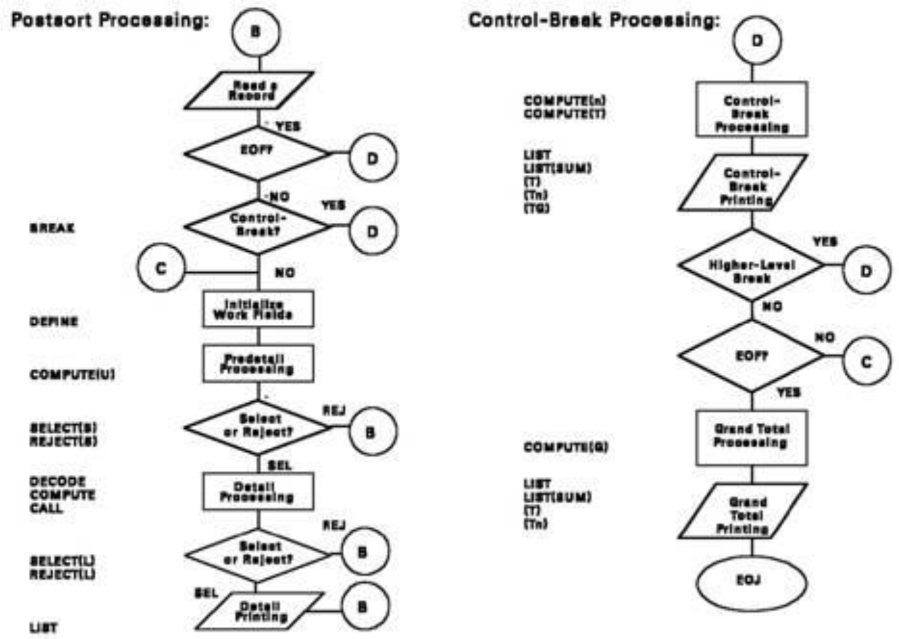
Appendix A: Logic Flowcharts

The CA JARS Wizard flow consists of the following:

- Initial Processing
- PreSORT Processing
- PostSORT Processing
- Control Break Processing

The flowcharts on the next two pages show the normal sequence of operations for a CA JARS Wizard run. They do not include special processing functions, such as RANK, EXCEPTION, CALENDAR, PLOT, and so forth.





This section contains the following topics:

Initial Processing

CA JARS Wizard reads and edits the control statements for the run. It lists the statements and prints diagnostic messages when necessary to call attention to possible problems.

CA JARS Wizard is a one-pass compiler. It checks for a glossary in the INPUT statement. It validates data field names in the control statements by looking at the field entries in the glossary. If the data field name is not in the glossary, you must ensure that a DECODE, DEFINE, or COMPUTE statement defining the new data field name occurs physically in the control statements ahead of the statement that refers to the data field name. If there are errors, a message is printed noting their location and the program is cancelled. Correct the errors listed and start again from the beginning.

If there are no errors, CA JARS Wizard prints its compilation and cross-reference listings, depending on the PARAMETER statement specifics, and proceeds to process the input file.

PreSORT Processing

If there is a SORT statement, CA JARS Wizard performs the following steps:

- Reads a record from the input file at the end of the file, and passes control to the SORT.
- Any CALL(I) statements are processed.
- Evaluates any COMPUTE(S) statements.
- Reset evaluates the SELECT(P) and REJECT(P) statements, and also the SELECT and REJECT statements that physically appear in the job stream ahead of the CALL(P), COMPUTE(P), and DECODE(P) statements.

  If the SELECT conditions are met and the REJECT conditions are not met, the record is processed further.

- If there are no SELECT or REJECT statements at this point, all of the records are processed further.
- The selected and nonrejected records are processed by executing the CALL(P), DECODE(P), and COMPUTE(P) statements.

  There can be multiple specifications for each of these control statements. All the statements that are of the same function execute the series of instructions as if they were one statement. The strung statements execute in the chronological order in which they appear in the input job stream.

  New fields generated as a result of these statements are appended to the input record. For example, if a COMPUTE(P) statement is executed and generates a 4-byte binary result called CTR, the input record length is extended by 4 bytes, and the last four bytes are accessed at any time, by reference to the field name CTR.

  Thus, all newly created fields at this point can be used as operands in succeeding statements.
- Evaluates the SELECT and REJECT statements that physically appear in the input job stream after the CALL(P), COMPUTE(P), and DECODE(P) statements.

  If the SELECT conditions are met and the REJECT conditions are not met, the record is processed further.

  If there are no SELECT or REJECT statements at this point, all of the records are processed further.
- The selected and nonrejected records are passed on to the SORT.
- At the end of the file, control is passed to the SORT program.

PostSORT Processing

If there is a SORT statement, CA JARS Wizard performs the following steps:

- Reads a record from the file of sorted records. If there is no SORT, it reads directly from the input file.
- Resets all the work fields to zero or to their initial values in the DEFINE statement, except for the work fields that were defined with the DEFINE(K) statement.
- Executes the COMPUTE(U) statements.
- Evaluates the SELECT(S) and REJECT(S) statements.

  If the SELECT(S) conditions are met or the REJECT(S) conditions are not met, the record is processed further.
- Performs the DECODE, COMPUTE, EXCEPTION, and CALL statements.

  There can be multiple statements for each of these control functions. CA JARS Wizard strings all the statements that are of the same function and executes the series of instructions as if they were one statement. The strung instructions execute in the chronological order in which they appear in the input job stream.

- Calculations for any CALENDAR report are performed.
- Evaluates the SELECT(L) and REJECT(L) statements. If the SELECT(L) conditions are met and the REJECT(L) conditions are not met, then the record is processed further.
- Prints the detail lines as specified in the LIST statement.
- At the end of the file, CA JARS Wizard loops through the steps above and after printing the highest level control break, it executes the series of COMPUTE(G) statements. It prints the horizontal grand total line as defined by the LIST or LIST(SUM) statement and the vertical grand total line as specified by the suffixes (T), (Tn) and (TG). Finally, it prints the run statistics and terminates.

Control Break Processing

At the occurrence of a control break, CA JARS Wizard performs the following steps:

- Executes the lowest level COMPUTE(n) statement where n is a value from 1 to 8, representing the 8 allowable control breaks.

  It then executes the COMPUTE(T) statements, prints the horizontal break lines specified by the LIST or LIST(SUM) statement, and prints the vertical total lines as specified by the suffixes (T) and (Tn), where n is equal to the control break level or less. Again, there may be multiple statements for each level of these control functions. CA JARS Wizard strings the conditions as if they are one statement and executes the instruction in the chronological sequence in which they appear in the input job stream.
- Evaluates if there are higher level control breaks and performs the above series with n + 1.

Summary of Statement Processing Sequence

If the CA JARS Wizard run does **not** use a SORT control statement:

(Reset Work Items)

CALL(I)

COMPUTE(S)

SELECT

REJECT

DECODE

COMPUTE

CALL

SELECT(L)

REJECT(L)

If the CA JARS Wizard run contains a SORT control statement:

PreSORT:

(Reset PreSORT Work Items)

CALL(I)

COMPUTE(S)

SELECT(P)/SELECT

REJECT(P)/REJECT

CALL(P)

DECODE(P)

COMPUTE(P)

SELECT

REJECT

PostSORT:

(Reset PostSORT Work Items)

COMPUTE(U)

SELECT(S)/SELECT

REJECT(S)/REJECT

DECODE

COMPUTE

CALL

SELECT(L)

REJECT(L)

Appendix B: Summary of Suffixes, Reserved Words and Keywords

This appendix describes all of the data field suffix codes and reserved words in CA JARS Wizard.

This section contains the following topics:

Suffix Codes

Suffix codes can be divided into four basic groups:

Control group suffixes

codes that affect a control group.

Data field suffixes

codes affecting the initial definition of all fields.

Date and time suffixes

codes that affect date and time fields. Suffixes for date fields affect the internal format of dates; time field suffixes affect the way times are printed.

Print control suffixes

codes that affect the printing of a field.

Several suffix codes are included in more than one of these groups. Tables of each of these groups are provided on the following pages. Each table describes the suffix codes associated with a specific group, and lists the control statements that the suffix codes can be used in.

You can append multiple suffix codes to a field name in a CA JARS Wizard run, such as field name(a)(b)(c), as long as the suffix codes do not contradict each other. Suffixes must be appended to the field name or to each other without intervening blanks.

The following suffix codes are mutually exclusive: (AV), (CT), (DF), (DL), (MN), (MX), (PC), (SD), and (TL). They cannot be specified together in an individual reference for a particular field. They may, however, be specified in different references for the same field.

Control Group Suffix Codes

Suffix Code	Description	Used with Statements
(-)	Causes a line of dashes to be printed after each line of group totals for the field.	BREAK
(AV)	Computes the average value of the field for all records in the control group. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (CT), (DF), (DL), (MN), (MX), (PC), (SD), and (TL).	COMPUTE LIST
(D)	The field is sorted in descending sequence.	BREAK SORT
(MN)	Computes the minimum value of the field for all records in the control group. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MX), (PC), (SD), and (TL).	COMPUTE LIST
(MX)	Computes the maximum value of the field for all records in the control group. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MN), (PC), (SD), and (TL).	COMPUTE LIST
(PC)	Computes the record or control group's percentage of the field total. PreSORT totalling is required for successful operation of this suffix code. This suffix does not work with fields defined on a LIST statement that is part of a RANK report, nor does it work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MN), (MX), (SD), and (TL).	COMPUTE LIST

Suffix Code	Description	Used with Statements
(R)	Causes the control group heading to be right-justified.	BREAK
(Sn)	Prints n blank lines after each line of control group totals for the field.	BREAK
(SB)	Suppresses printing totals for this break if there is only one break at the lower level, or in the case of the lower level break, only one record.	BREAK
(SD)	Computes the standard deviation of the field for all records in the control group. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MN), (MX), (PC), and (TL).	COMPUTE LIST
(SK)	Skips to a new page after printing each line of control group totals for a field.	BREAK
(T)	Requests vertical totals for a field after every control group break. The field does not appear in the detail print line.	LIST
(Tn)	Requests vertical totals for a field after control group breaks of n level and higher. n must have a value from 1 to 8.	LIST

Data Field Suffix Codes

Suffix Code	Description	Used with Statements
(.n)	Indicates the field has n decimal positions, or if the field is hexadecimal, defines the bit position. When used with a field in a LIST statement, the printed data for the field is truncated to the specified decimal position.	COMPUTE DECODE DEFINE LIST

Suffix Code	Description	Used with Statements
(n)	Specifies the length of a field (n). Valid values for n are: 1-16 for numeric fields; 1-255 for character fields.	COMPUTE DECODE DEFINE
(x,y)	Used to access a subfield within a character field. Specify x as the first position of the subfield and y as the number of positions. This suffix must not be used with numeric or date fields.	ANY
(A)	Ranks records in ascending order by the specified field.	RANK
(AV)	Computes the average value of the field for all records in the control group. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (CT), (DF), (DL), (MN), (MX), (PC), (SD), and (TL).	COMPUTE LIST
(B)	Specifies that a field contains binary data only.	COMPUTE DECODE DEFINE
(C)	Specifies that a field contains EBCDIC character data only.	COMPUTE DECODE DEFINE
(D)	The field is sorted in descending sequence.	BREAK SORT
(DD/MM/YY)	Specifies that a date field must be a character field with an internal format of DD/MM/YY.	COMPUTE DECODE DEFINE
(DDMMYY)	Specifies that a date field must be a numeric, packed, or character field with an internal format of DDMMYY.	COMPUTE DECODE DEFINE
(Gn)	The field prints in plot format using its default specifications or as described in a corresponding PLOT Gn statement. Specify n from 1 to 6.	LIST
(H)	Specifies that a field contains hexadecimal data only.	COMPUTE DECODE DEFINE

Suffix Code	Description	Used with Statements
(K)	Indicates that a field is maintained by the user CA JARS Wizard does not automatically reset it.	DEFINE COMPUTE
(MM/DD/YY)	Specifies that a date field must be a character field with an internal format of MM/DD/YY.	COMPUTE DECODE DEFINE
(MMDDYY)	Specifies that a date field must be a numeric, packed, or character field with an internal format of MMDDYY.	COMPUTE DECODE DEFINE
(MN)	Computes the minimum value of the field for all records in the control group. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MX), (PC), (SD), and (TL).	COMPUTE LIST
(MX)	Computes the maximum value of the field for all records in the control group. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MN), (PC), (SD), and (TL).	COMPUTE LIST
(N)	Specifies that a field contains numeric, zoned decimal data only.	COMPUTE DECODE DEFINE
(NOSP)	Indicates that the field cannot be split into time slices by the Interval Chopper. This is the default for all fields.	COMPUTE DECODE DEFINE
(P)	Specifies that a field contains numeric, packed decimal data only.	COMPUTE DECODE DEFINE

Suffix Code	Description	Used with Statements
(PC)	Computes the record or control group's percentage of the field total. PreSORT totalling is required for successful operation of this suffix code. This suffix does not work with fields defined on a LIST statement that is part of a RANK report, nor does it work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MN), (MX), (SD), and (TL).	COMPUTE LIST
(SD)	Computes the standard deviation of the field for all records in the control group. This suffix does work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MN), (MX), (PC), and (TL).	COMPUTE LIST
(SP)	Indicates that the field can be split into time slices by the Interval Chopper. Any user-created fields MUST have an (SP) suffix to be printed in an Interval Chopper report.	COMPUTE DECODE DEFINE
(ST)	Resets print options for the specified time field. All time values for this field are expressed in seconds.	COMPUTE DECODE DEFINE LIST
(TL)	Computes the total value of the field. PreSORT totalling is required for successful operation of this suffix code. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with (AV), (CT), (DF), (DL), (MN), (MX), (PC), and (SD).	COMPUTE LIST
(YY/MM/DD)	Specifies that a date field must be a character field with an internal format of YY/MM/DD.	COMPUTE DECODE DEFINE

Suffix Code	Description	Used with Statements
(YYDDD)	Specifies that a date field must be a numeric, packed or character field with an internal format of YYDDD (Julian date).	COMPUTE DECODE DEFINE
(YYMMDD)	Specifies that a date field must be a numeric, packed, or character field with an internal format of YYMMDD.	COMPUTE DECODE DEFINE

Date and Time Field Suffix Codes

Suffix Code	Description	Used with Statements
(DD/MM/YY)	Specifies that a date field must be a character field with an internal format of DD/MM/YY.	DECODE DEFINE
(DDMMYY)	Specifies that a date field must be a numeric, packed, or character field with an internal format of DDMMYY.	COMPUTE DECODE DEFINE
(HH)	The time field is converted from seconds to display as hours and hundredths of an hour (HH.HH).	COMPUTE DECODE DEFINE LIST
(HM)	The time field is converted from seconds to display as hours and minutes (HH:MM).	COMPUTE DECODE DEFINE LIST
(HS)	The time field is converted from seconds to display as hours, minutes, and seconds (HH:MM:SS).	COMPUTE DECODE DEFINE LIST
(KH)	The time field is converted from seconds to display as hours, minutes, seconds, and milliseconds (HH:MM:SS.SSS).	COMPUTE DECODE DEFINE LIST

Suffix Code	Description	Used with Statements
(KM)	The time field is converted from seconds to display as minutes, seconds, and milliseconds (MM:SS.SSS).	COMPUTE DECODE DEFINE LIST
(KS)	The time field is converted from seconds to display as seconds and milliseconds (SS.SSS).	COMPUTE DECODE DEFINE LIST
(MM/DD/YY)	Specifies that a date field must be a character field with an internal format of MM/DD/YY.	COMPUTE DECODE DEFINE
(MMDDYY)	Specifies that a date field must be a numeric, packed, or character field with an internal format of MMDDYY.	COMPUTE DECODE DEFINE
(MS)	The time field is converted from seconds to display as minutes and seconds (MM:SS).	COMPUTE DECODE DEFINE LIST
(ST)	Resets print options for the specified time field. All time values for this field are expressed in seconds.	COMPUTE DECODE DEFINE LIST
(YY/MM/DD)	Specifies that a date field must be a character field with an internal format of YY/MM/DD.	COMPUTE DECODE DEFINE
(YYDDD)	Specifies that a date field must be a numeric, packed or character field with an internal format of YYDDD (Julian date).	COMPUTE DECODE DEFINE
(YYMMDD)	Specifies that a date field must be a numeric, packed, or character field with an internal format of YYMMDD.	COMPUTE DECODE DEFINE

Print Control Suffix Codes

Suffix Code	Description	Used with Statements
(.n)	Indicates the field has n decimal positions, or if the field is hexadecimal, defines the bit position. When used with a field in a LIST statement, the printed data for the field is truncated to the specified decimal position.	COMPUTE DECODE DEFINE LIST
($)	The field prints with a dollar sign before the first significant digit.	COMPUTE DECODE DEFINE LIST
(-)	Causes a line of dashes to be printed after each line of group totals for the field.	BREAK
(Cn)	Sets a column number for a vertical totals item. The value specified for n must be from 1 to 6. Each number (1-6) represents 1/6 of the print area available, from left to right on the page.	LIST
(CT)	Forces printing of a nonnumeric field on the total line for a control group. The data displayed as the total for this nonnumeric field is taken from the first record in the control group. (CT) is mutually exclusive with (AV), (DF), (DL), (MN), (MX), (PC), (SD), and (TL).	COMPUTE DECODE DEFINE LIST
(DC)	Prints the contents of character fields centered within the field.	COMPUTE DECODE DEFINE LIST
(DF)	Forces printing of a nonnumeric field on the total line for a control group. The data displayed as the total of this nonnumeric field is taken from the first record in the control group. (DF) is mutually exclusive with (AV), (CT), (DL), (MN), (MX), (PC), (SD), and (TL).	COMPUTE DECODE DEFINE LIST

Suffix Code	Description	Used with Statements
(DL)	Prints the value of a field from the last record in a control group on the control group total line. This suffix does not work with numeric fields defined as (NV). It is also mutually exclusive with: (AV), (CT), (DF), (MN), (MX), (PC), (SD), and (TL).	LIST
(F)	Indicates that field values for the specified field are ordered in decreasing total order, instead of alphabetic or user code order.	XTAB
(Fcn)	Supplies the fill character specifications for the fanning of vertical totals.	LIST
(Gn)	The field prints in plot format using its default specifications or as described in a corresponding PLOT Gn statement. Specify n from 1 to 6.	LIST
(HH)	The time field is converted from seconds to display as hours and hundredths of an hour (HH.HH).	COMPUTE DECODE DEFINE LIST
(HM)	The time field is converted from seconds to display as hours and minutes (HH:MM).	COMPUTE DECODE DEFINE LIST
(HS)	The time field is converted from seconds to display as hours, minutes, and seconds (HH:MM:SS).	COMPUTE DECODE DEFINE LIST
(KH)	The time field is converted from seconds to display as hours, minutes, seconds, and milliseconds (HH:MM:SS.SSS).	COMPUTE DECODE DEFINE LIST
(KM)	The time field is converted from seconds to display as minutes, seconds, and milliseconds (MM:SS.SSS).	COMPUTE DECODE DEFINE LIST

Suffix Code	Description	Used with Statements
(KS)	The time field is converted from seconds to display as seconds and milliseconds (SS.SSS).	COMPUTE DECODE DEFINE LIST
(Ln)	Reserves n print positions for a data field, overriding its defined length. This suffix, is intended primarily for numeric fields; it should not be used to truncate character fields. For that purpose the (x,y) data field suffix is recommended instead.	COMPUTE DECODE DEFINE LIST
(LA)	Prints the contents of character fields left-aligned within the field. This is the default alignment for character fields when they are printed.	COMPUTE DECODE DEFINE LIST
(MS)	The time field is converted from seconds to display as minutes and seconds (MM:SS).	COMPUTE DECODE DEFINE LIST
(NA)	Suppresses printing an average for a field when averaging has been requested for all numeric fields by a PARAMETER AVERAGE statement.	COMPUTE DECODE DEFINE LIST
(ND)	Suppresses printing of numeric data in detail lines. Numeric data is only printed in total lines.	COMPUTE DECODE DEFINE LIST
(NG)	Suppresses printing of a grand total for this item.	COMPUTE DECODE DEFINE LIST
(NP)	Suppresses all printing of the field. (Not to be confused with NP command.)	
(NR)	Suppresses printing of a field if the value of the field is the same in this line as it was in the previous line.	COMPUTE DECODE DEFINE LIST

Suffix Code	Description	Used with Statements
(NT)	Suppresses printing of totals for this item.	COMPUTE DECODE DEFINE LIST
(NU)	Suppresses printing of the underscore character before printing totals.	COMPUTE DECODE DEFINE LIST
(NV)	Suppresses automatic summarization and editing (addition of leading zeros, commas, and decimal points) of numeric fields. Eliminates all use of computational suffix codes: (AV), (DL), (MN), (MX), (PC), (SD), and (TL).	COMPUTE DECODE DEFINE LIST
(R)	Causes the control group heading to be right-justified.	BREAK
(RA)	Prints the contents of character fields right-justified within the field.	COMPUTE DECODE DEFINE LIST
(Sn)	When used with LIST, (Sn) specifies a column number for a vertical totals item. The column number indicates what column the value should be printed in when the value is to be printed on the same line as the previous vertical totals item. The value specified for n must be from 1 to 6. Each number (1-6) represents 1/6 of the print area available, from left to right on the page.	LIST
	When used with BREAK, (Sn) prints n blank lines after each line of control group totals for the field.	BREAK
(SK)	Skips to a new page after printing each line of control group totals for a field.	BREAK

Suffix Code	Description	Used with Statements
(ST)	Resets print options for the specified time field. All time values for this field are expressed in seconds.	COMPUTE DECODE DEFINE LIST
(T)	Requests vertical totals for a field after every control group break. The field does not appear in the detail print line.	LIST
(Tn)	Requests vertical totals for a field after control group breaks of n level and higher. n must have a value from 1 to 8.	LIST
(TG)	Requests vertical totals for a field at the end of the report and after printing grand totals. The vertical totals do not appear elsewhere in the report.	LIST
(ZE)	Suppresses the printing of data fields with zero values.	COMPUTE DECODE DEFINE LIST

Reserved Words

The following table lists all of the reserved words in CA JARS Wizard.

Reserved Word	Description
ALWAYS	Used in a conditional statement to specify instructions that must always be executed regardless of the outcome of the conditional test(s).
AND	Connects two or more conditional tests.
BLANK	Refers to a field of blank spaces. This can be used to assign a value of blanks to character fields.
ELSE	Used in a conditional statement to specify instructions that should only be executed if the outcome of the conditional test(s) is false.
EQ	The alphabetic representation of the comparison operator **equal to.**

Reserved Word	Description
GE	The alphabetic representation of the comparison operator **greater than or equal to.**
GO TO	Transfers control to another set of instructions using a predetermined tag in a COMPUTE statement.
GT	The alphabetic representation of the comparison operator **greater than.**
IF	Begins all conditional statements and is usually followed by one or more conditional tests.
LE	The alphabetic representation of the comparison operator **less than or equal to.**
LOVALUE	May be used to assign a value of zero to a numeric field, or blanks to a character field.
LT	The alphabetic representation of the comparison operator **less than.**
MOVE	Transfers data from one field to another in a COMPUTE statement.
NE	The alphabetic representation of the comparison operator **not equal to.**
NEW	Used in COMPUTE, SELECT, and REJECT statements to test whether consecutive records contain different values for a specified field.
NG	The alphabetic representation of the comparison operator **not greater than.**
NL	The alphabetic representation of the comparison operator **not less than.**
NP	Forces a page break in reports.
OFF	Used to test individual bits in a hexadecimal field.
ON	Used to test individual bits in a hexadecimal field.
RANK	An internal variable used in ranked record reports.
SAME	Used in COMPUTE, SELECT, and REJECT statements to test whether consecutive records contain identical values for a specified field.
SORTKEY	An internal variable used in exception reports.
SPACE	Refers to a field of blank spaces. This can be used to assign a value of blanks to character fields.

Reserved Word	Description
THEN	Used in a conditional statement to specify instructions that should only be executed if the outcome of the conditional test(s) is true.
TO	Used when transferring data from one field to another.

Keywords

This product generates values for certain keywords. Use these keywords in CA JARS Wizard processing statements. It is not necessary to define the fields. Insert two blanks or another keyword after a literal in a statement that contains more than one keyword. These keywords are displayed in the following table.

Keyword	Description
CALENDAR DATE	The date field used for a calendar report. If this field is not defined, then the glossary field named DATE is used.
COMPANY	The default company ID (CA Inc.).
COUNT	The number of records read. This value is one for the first record and is incremented by one for each new record processed. When used in a control level break, the value is the number of records processed for the control group.
DAY OF WEEK	A value from 1 through 7, where one indicates Monday, two indicates Tuesday, etc. CA JARS Wizard computes this value from the DATE fields of the record.
GENLEVEL	The current genlevel.
GLOSSARY	The name of the primary input file.
HIDATE	The highest date value encountered in the DATE field of the records.
HITIME	The highest time value encountered in the START TIME field of the records with the highest DATE field value.
INPUT RECORD NO	The position of the record in the file - preSORT. The position of the record in the sorted file - postSORT.
LIMIT	The number of records read before an end-of-file is forced. (This can be used with COMPUTE statements only.)
LODATE	The lowest date value encountered in the DATE field of the records.

Keyword	Description
LOTIME	The lowest time value encountered in the START TIME field of records with the lowest DATE field value.
NO OF RECORDS	The number of selected records either preSORT or postSORT.
RELEASE	The current release.
TOD	The time of day in seconds.
TODAY	The current date and time of day. The date is formatted as: DD MMM YYYY.
WEEK NO	A week number is assigned, depending on the date. A week begins every Monday.
WEEKDAY	The DATE field is converted to the name of the day, that is, Monday, Tuesday, etc.
WIZPAGES	The current page number of the report.

Appendix C: WIZIVP: The Interval Chopper

The CA JARS Wizard Interval Chopper is a special facility that allows you to create time analysis reports on your input data. Records from your input file are grouped into user-specified time intervals. As each record is added to an interval, its numeric fields are summed with the numeric fields of the other records in that interval. Thus, each time interval record is a single summary record of all of the records from your input file that belong to the time interval. These time interval records are then used to create your output.

Note: The Interval Chopper can be used with the z/OS Combined Billing Facility (CBF) glossary.

This section contains the following topics:

Operations

To execute the Interval Chopper you must:

Include a CALL statement for the Interval Chopper program, WIZIVP, in the control statements for the run. For example:

```
CALL WIZIVP
```

Define a value for the Interval Chopper field INTERVAL. INTERVAL is used by the Interval Chopper to specify the reporting interval. If no value is specified, CA JARS Wizard assumes a default of 15 minutes. Use the following format when defining INTERVAL:

```
DEFINE(K) INTERVAL EQ n
```

The (K) suffix code is appended to the DEFINE statement because INTERVAL is a constant field that is maintained strictly by you. Specify n as a value from 1 to 60 minutes. The value of n must be evenly divisible into 60. For example,

```
DEFINE(K) INTERVAL EQ 12
```

This defines the INTERVAL as 12 minutes. Notice that 12 divides evenly into 60 (5 times); there is no remainder.

Once the value of INTERVAL has been defined, CA JARS Wizard reads all the records in the input file and adds them to the appropriate time interval record. If an input file record extends over more than one time interval, multiple segments of the input record are created to allocate the input record statistics to your user-defined time interval records.

For each time interval record generated, two fields are created: INTERVAL START and INTERVAL LENGTH. These contain the starting time of the record (INTERVAL START) and the length of the record (INTERVAL LENGTH). Each of these fields can be used for sorting and breaking operations. Time analysis reports can then be produced.

Data Field Requirements

Assign (SP) and (NOSP) suffix codes to fields to indicate whether they can be included in the time interval records.

(SP)

Indicates that the field can be used by the Interval Chopper and split into time slices, as necessary.Any user-created work fields must have an (SP) suffix to be printed in an Interval Chopper report.

(NOSP)

Indicates that the field cannot be used by the Interval Chopper.This is the default for all fields.

(SP) and (NOSP) suffix codes can be assigned to fields either in the product glossary or when they are created. If you do not specify (SP) for a user-created work field, (NOSP) is assumed.

Both glossary and work fields can be included in a time interval record provided they meet the following conditions:

- They must have packed, binary, or numeric formats.
- An (SP) suffix code must be assigned to the field.
- The field must not be a date field. Date fields can never be included in a time interval record.
- Work fields must be calculated prior to the execution of the Interval Chopper required CALL WIZIVP control statement.

Sample Usage

The Interval Chopper is a CA JARS Wizard function that reads the input records and chops them into one or more records in order to allocate the statistics in an input record to a time interval record. You define the time interval using a DEFINE(K) control statement for the Interval Chopper field, INTERVAL. As the following example shows, a report on time sequence can be produced, providing on-the-hour analysis of system activity.

```
INPUT primary-input-file
SELECT REC/TYPE EQ 'E'
DEFINE(K) INTERVAL EQ 60
CALL(P) WIZIVP
SORT
BREAK INTERVAL START
COMPUTE IF INTERVAL START GT 86400 THEN TIME(K)(HM) EQ
        INTERVAL START - 86400 ELSE
        TIME(NG) EQ INTERVAL START
LIST(SUM) TIME(L5)(NG), E.NO OF JOBS, ELAPS TIME, E.ACTIVE TIME,
          E.CPU TIME(HM)
```

Appendix D: CAWGJR70 Glossary

Two tables have been provided to help you use the CAWGJR70 glossary. The first table gives the record layout of the glossary and a brief description of each field in the glossary record. The second table shows CA JARS data elements and their corresponding CAWGJR70 glossary item. You can use this second table to convert your CA JARS reports to CA JARS Wizard reports.

Note: Only use CA JARS history or summary files as input to CA JARS Wizard. Do not use the CA JARS account file.

This section contains the following topics:

CAWGJR70 Glossary Record Layout

Name	Pos	Len	Format	Suffixes	Description
CAWGJR70/RECORD	1	612	H		JARS LAYOUT W/CA JARS Wizard
RDW	1	4	B	NV	RECORD DESCRIPTOR WORD
CPU/ID	5	1	C		CPU IDENTIFI- CATION
FILE/LEVEL	6	1	C		FILE LEVEL
RCD/IND	7	1	P	NV	RECORD INDICATOR
JOB/GROUP	8	20	C		UNIQUE JOB KEY
RDR STRT/DTE/YMD	8	6	C	YYMMDD	READER START DATE (YYMMDD)
DATE	8	6	C	YYMMDD	(YYMMDD)
RDR STRT/DTE/YY	8	2	C	NV	READER START DATE YEAR
RDR STRT/DTE/MM	10	2	C	NV	READER START DATE MONTH
RDR STRT/DTE/DD	12	2	C	NV	READER START DATE DAY
RDR STRT/TME/HMS	14	6	C	NV	READER START TIME (HHMMSS)
RDR STRT/TME/HH	14	2	C	NV	READER START TIME HOUR
RDR STRT/TME/MM	16	2	C	NV	READER START TIME MINUTE

Name	Pos	Len	Format	Suffixes	Description
RDR STRT/TME/SS	18	2	C	NV	READER START TIME SECOND
JOB/NAME	20	8	C		
RJE LINE/NAME	20	8	C		
STEP/JOB/IND	28	2	C		STEP/JOB INDICATOR 1B STEP RECORD
STEP/NBR	30	2	B		NUMBER OF STEPS AT JOB, SUMMARY LEVELS
PRO/ID	32	1	C		PROCESSING ID
USER/ID	33	8	C		BLANK UNLESS FILLED BY SMF USER EXIT
START/TIME/HMS	41	6	C	NV	START TIME (HHMMSS)
CONV/START/TIME	41	6	C	NV	APPC CONVERSATION START TIME (HHMMSS)
TRAN/START/TIME	41	6	C	NV	APPC TRANSACTION START TIME (HHMMSS)
START/TIME/HH	41	2	C	NV	START TIME HOUR
START/TIME/MM	43	2	C	NV	START TIME MINUTE
START/TIME/SS	45	2	C	NV	START TIME SECOND
RUN/DATE/YMD	47	6	C	YYMMDD	RUN DATE (YYMMDD)
RUN/DATE/YY	47	2	C	NV	RUN DATE YEAR
RUN/DATE/MM	49	2	C	NV	RUN DATE MONTH
RUN/DATE/DD	51	2	C	NV	RUN DATE DAY
CL	53	2	C		CLASS
STOP/TIME/HMS	55	6	C	NV	STOP TIME (HHMMSS)
CONV/STOP/TIME	55	6	C	NV	APPC CONVERSATION STOP TIME (HHMMSS)
TRAN/STOP/TIME	55	6	C	NV	APPC TRANSACTION STOP TIME (HHMMSS)
STOP/TIME/HH	55	2	C	NV	STOP TIME HOUR
STOP/TIME/MM	57	2	C	NV	STOP TIME MINUTE
STOP/TIME/SS	59	2	C	NV	STOP TIME SECOND
PROGRAMMER	61	20	C		FROM OS JOB STATEMENT
CONV ID	63	18	C		APPC CONVERSATION ID

Name	Pos	Len	Format	Suffixes	Description
USER INFO	81	16	C		
ACCT FLDS	81	16	C		
CC	97	4	C		COMPLETION/ CANCEL CODE
TI	101	2	C		TERMINATION INDICATOR
STEP/NAME	103	8	C		
REMOTE/NAME	103	8	C		
PROGRAM/NAME	111	8	C		
ORIGINAL/TP NAME	111	8	C		APPC ORIGINAL TP NAME
RJE/LINE/PASSWRD	111	8	C		
CORE/ALOC	119	4	P	L5 MX	1K INCREMENTS
CORE/USED	123	4	P	L5 MX	1K INCREMENTS
PCT/K	127	3	P	L3 MX	99999C; PERCENT OF UNUSED CORE
JOB/NBR	130	5	C		
TSU/NBR	130	5	C		
P/R	135	2	C		PRIORITY
CARDS/READ	137	6	P	L9	
LINES/PRINTED	143	6	P	L10	
SPCL LINES/PRTD	149	6	P	L10	
CARDS/PUNCHED	155	6	P	L9	
PRNT/FORM	161	4	C		
PNCH/FORM	165	4	C		
IDLE/TIME/HRS	169	6	P	.2 L10	999999V99999C; HOURS
ELAPSED/TIME/HRS	175	6	P	.2 L10	999999V99999C; HOURS
CPU/TIME/HRS	181	6	P	.2 L10	999999V99999C; HOURS
OVHEAD/TIME/HRS	187	6	P	.2 L10	999999V99999C; HOURS; OVERHEAD TIME
ACT/TIME/RJE/HRS	187	6	P	.2 L10	999999V99999C; HOURS; ACTIVE RJE TIME
TCB/TIME	187	6	P	.2 L10	999999V99999C HOURS APPC TCB TIME

Name	Pos	Len	Format	Suffixes	Description
WAIT/TIME/HRS	193	6	P	.2 L10	999999V99999C; HOURS
CNT/TIME/RJE/HRS	193	6	P	.2 L10	999999V99999C; HOURS; CONNECT RJE TIME
SRB/TIME	193	6	P	.2 L10	999999V99999C; HOURS; APPC SRB TIME
RDR Q/TIME/HMS	199	6	N	NV	READER QUEUE TIME AVERAGE (HHMMSS)
RDR Q/TIME/HH	199	2	N	NV	READER QUEUE TIME HOURS
RDR Q/TIME/MM	201	2	N	NV	READER QUEUE TIME MINUTES
RDR Q/TIME/SS	203	2	N	NV	READER QUEUE TIME SECONDS
WTR Q/TIME/HMS	205	6	N	NV	WRITER QUEUE TIME AVERAGE (HHMMSS)
WTR Q/TIME/HH	205	2	N	NV	WRITER QUEUE TIME HOURS
WTR Q/TIME/MM	207	2	N	NV	WRITER QUEUE TIME MINUTES
WTR Q/TIME/SS	209	2	N	NV	WRITER QUEUE TIME SECONDS
TAROUND/TIME/ HMS	211	6	N	NV	TURNAROUND TIME AVERAGE (HHMMSS)
TAROUND/TIME/HH	211	2	N	NV	TURNAROUND TIME HOURS
TAROUND/TIME/ MM	213	2	N	NV	TURNAROUND TIME MINUTES
TAROUND/TIME/SS	215	2	N	NV	TURNAROUND TIME SECONDS
READER/IO COUNT	217	7	P	L10	
RJE LINE/EXCP CT	217	7	P	L10	
BYTES/SEND	217	7	P	L10	APPC BYTES SENT
PRINTER/IO COUNT	224	7	P	L10	
RJE LINE/NAK CT	224	7	P	L10	
BYTES/REC	224	7	P	L10	APPC BYTES RECEIVED
PUNCH/IO COUNT	231	7	P	L10	
RJE LINE/DATA CK	231	7	P	L10	
NO OF/CALLS	231	7	P	L10	APPC NUMBER OF CALLS
TAPE/IO COUNT	238	7	P	L10	
RJE/INV LG	238	7	P	L10	

Name	Pos	Len	Format	Suffixes	Description
NO OF/CONV	238	7	P	L10	APPC NUMBER OF CONVERSATIONS
DISK/IO COUNT	245	7	P	L10	
RJE LINE/ERRORS	245	7	P	L10	
EXCP/COUNT	245	7	P	L10	APPC EXCP COUNT
OTHER/IO COUNT	252	7	P	L10	
RJE LINE/TME OUT	252	7	P	L10	
DEV/CONN/TIME	252	7	P	L10	9999999V999 SECS; APPC DEV CONNECT
TOTAL/IO COUNT	259	7	P	L10	
IO INDX	266	4	P	NV	
S/I	270	1	C		STORAGE INDICATOR V=VIRTUAL
APPC/TYPE	270	1	C		APPC TYPE:
PAGE IN/COUNT	271	7	P	L10	
PAGE OUT/COUNT	278	7	P	L10	
TOTAL/PAGES	285	7	P	L10	
CPU/PAGE	292	4	P	L4 NV	
ELAP/PAGE	296	4	P	L4 NV	
G/1	300	1	C		GROUP CODE 1
G/2	301	1	C		GROUP CODE 2
G/3	302	1	C		GROUP CODE 3
P/N	303	2	C		
JOB/RJE/TYPE	303	2	C		JOB/RJE TYPE:
INPUT/DEVICE	305	8	C		
LOCAL/LU NAME	305	8	C		APPC LOCAL LU NAME
RDR/DURATION/ HRS	313	6	P	L10	999999V99999C; HOURS
WTR/DURATION/ HRS	319	6	P	L10	999999V99999C; HOURS
SWAP/COUNT	325	7	P	L10	
SWAP/PAGES IN	332	7	P	L10	
SWAP/PAGES OUT	339	7	P	L10	

Name	Pos	Len	Format	Suffixes	Description
TOTAL SWAP/PAGES	346	7	P	L10	
NO/RD	353	2	P	L2	# OF READERS USED
NO/PR	355	2	P	L2	# OF PRINTERS USED
NO/PU	357	2	P	L2	# OF PUNCHES USED
NO/TP	359	2	P	L2	# OF TAPES USED
NO/DK	361	2	P	L2	# OF DISKS USED
NO/OT	363	2	P	L2	# OF OTHERS USED
NO/DU	365	2	P	L2	# OF DISKS - PRIVATE USED
ND	367	2	P	L2	# OF DEVICES USED
TSO/TPUTS	369	4	P	L7	TSO ONLY
TSO/TGETS	373	4	P	L7	TSO ONLY
ACTIVE/TIME/HRS	377	6	P	L10	999999V99999C; HOURS
CONNECT/TIME/HRS	383	6	P	L10	999999V99999C; HOURS
ALLOC/TIME/HRS	389	6	P	L10	999999V99999C; HOURS; ALLOCATION TIME
RES/TIME/HRS	395	6	P	L10	999999V99999C; HOURS; RESIDENT TIME
SRB/TIME/HRS	401	6	P	L10	99999V99999C; HOURS
TCB/TIME/HRS	407	6	P	.2 L10	999999V99999C; HOURS
SERVICE/UNITS	413	6	P	L10	MVS ONLY
SERV/RATE	419	3	P	L5 NV	MVS ONLY
CPU/IDX	422	2	P	L3 NV	MVS ONLY
PAGE/SECONDS	424	6	P	L10	MVS ONLY
PER/GRP	430	3	C		MVS ONLY
INP/RTE	433	3	C		MVS ONLY
PRT/RTE	436	3	C		MVS ONLY
RQ/ID	439	2	C		JOB REQ
D/I	441	1	C		(1 = MONDAY 7 = SUNDAY)
G/4	442	1	C		
G/5	443	1	C		
G/6	444	1	C		

Name	Pos	Len	Format	Suffixes	Description
FORMS/COUNT	445	2	P	NV	
O/C	447	1	C		SYSOUT CLASS
O/T	448	1	C		SYSOUT TYPE
I/C	449	1	C		SYSOUT INTV CODE
OUTPUT/DVC NAME	450	8	C		
PARTNER/LU NAME	450	8	C		APPC PARTNER LU NAME
STD LINES/PRNTD	458	6	P	L10	
STD PAGES/PRNTD	464	5	P	L9	
SPCL PAGES/PRNTD	469	5	P	L9	
TOT PAGES/PRNTD	474	5	P	L9	
TAPE/MOUNTS	479	4	P	L6	
DISK/MOUNTS	483	4	P	L6	
ABSORP/RATE	487	4	P	L6 NV	
USER INFO2	491	16	C		
USER INFO3	507	16	C		
USER/FIELD	523	16	C		
USER/COUNT	539	5	P		
USER/TIME/HRS	544	5	P	L10	9999V99999; HOURS
AUDIT/FLAGS	549	6	C		SMF AUDIT FLAGS
PROCESS/TIME/HRS	555	6	P	L10	999999V99999C; HOURS
PROCESS/CHARGE	561	6	P	L11	999999999V99C; DOLLARS
IO/TIME/HRS	567	6	P	L10	999999V99999C; HOURS
IO/CHARGE	573	6	P	L11	999999999V99C; DOLLARS
UR/CHARGE	579	5	P	L9	9999999V99C; DOLLARS
SETUP/CHARGE	584	5	P	L10	9999999V99C; DOLLARS
TOTAL/CHARGE	589	6	P	L11	999999999V99C; DOLLARS
C/S	595	1	C		CHARGE SUFFIX
ADJUSTED/RATE	596	6	P	.2 L9 NV	999999999V99C; DOLLARS
CONNECT/CHARGE	602	5	P	.2 L9	9999999V99C; DOLLARS
RJE/CHARGE	607	5	P	.2 L9	9999999V99C; DOLLARS

Name	Pos	Len	Format	Suffixes	Description
RSRVD01	612	1	P	NV	

CA JARS Data Element - CAWGJR70 Conversion Table

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
01	Control Field	not applicable	
02	Description Field	not applicable	
03	Processing ID	PRO/ID	
04	Job Name	JOB/NAME	
05	Program/Phase Name	PROGRAM/NAME	
06	Start (Run) Date	RUN/DATE	To use RUN/DATE in a CA JARS Wizard run, insert copybook JRUNDATE in the run.
07	Job Class	CL	
08	Accounting Field/User Info. for VSE	USER/INFO	
09	Completion/Cancel Code	CC	
10	User Identification	USER/ID	
11	Step Name	STEP/NAME	
12	Start Time	START/TIME	To use START/TIME in a CA JARS Wizard run, insert copybook JSTRTIME in the run.
13	Stop Time	STOP/TIME	To use STOP/TIME in a CA JARS Wizard run, insert copybook JSTOPTME in the run.
14	Job Number	JOB/NBR	At summary level, use COUNT for number of records processed in the control group.
15	Priority Level	P/R	
16	Print Forms ID	PRNT/FORM	
17	Punch Forms ID	PNCH/FORM	
18	Cards Read	CARDS/READ	

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
19	Lines Printed (Total)	LINES/PRINTED	
20	Cards Punched	CARDS/PUNCHED	
21	Core Used	CORE/USED(MX)	Append the (MX) suffix to CORE/USED to get the actual equivalent of data element 21.
22	Core Allocated	CORE/ALOC(MX)	Append the (MX) suffix to CORE/ALOC to get the actual equivalent of data element 22.
23	Percent of Unused Core	PCT/K(MX)	Append the (MX) suffix to PCT/K to get the actual equivalent of data element 23.
24	Setup/Idle Time	IDLE/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 24: COMPUTE IDLE/TIME EQ (IDLE/TIME/HRS * 3600)
25	Elapsed Time-in hours	ELAPSED/TIME/ HRS	
26	CPU Time-in hours	CPU/TIME/HRS	
27	Overhead Time-in hours	OVHEAD/TIME/HRS	
28	Wait Time - in hours	WAIT/TIME/HRS	
29	Processor Time-in hours	PROCESS/TIME/ HRS	
30	I/O Time-in hours	IO/TIME/HRS	
31	Total Time-in hours	TOTAL/TIME/HRS	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 31: COMPUTE TOTAL/TIME/HRS EQ (PROCESS/TIME/ HRS + IO/TIME HRS)
32	Total # of Devices Used	ND	
33	Reader I/O Count	READER/IO COUNT	
34	Printer I/O Count	PRINTER/IO COUNT	
35	Punch I/O Count	PUNCH/IO COUNT	

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
36	U/R I/O Count	UR/IO COUNT	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 36: COMPUTE UR/IO COUNT EQ (READER/IO COUNT + PRINTER/IO COUNT + PUNCH/IO COUNT)
37	Tape I/O Count	TAPE/IO COUNT	
38	Disk I/O Count	DISK/IO COUNT	
39	Other I/O Count	OTHER/IO COUNT	
40	Total I/O Count	TOTAL/IO COUNT	
41	% Total Charge	TOTAL/CHARGE (PC)	Append the (PC) suffix to TOTAL/CHARGE to get the actual equivalent of data element 41.
43	Adjusted Rate	ADJUSTED/RATE	
44	Processor Charge	PROCESS/CHARGE	
45	I/O Charge	IO/CHARGE	
46	Total Charge	TOTAL/CHARGE	
47	Distributed Charge	not applicable	
54	U/R Charge	UR/CHARGE	
55	Setup Charge	SETUP/CHARGE	
56	Step Number	STEP/NBR	At summary level, use COUNT for number of records processed in the control group.
57	CPU Identification	CPU/ID	
58	Programmer Name	PROGRAMMER	
59	Termination Indicator	TI	
60	Reader Start Date	RDR STRT/DTE	To use RDR STRT/DTE in a CA JARS Wizard run, insert copybook JRDRDATE in the run.
61	Reader Start Time	RDR STRT/TME	To use RDR STRT/TME in col a CA JARS Wizard run, insert col copybook JRDRSTME in col the run.

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
62	Reader Queue Time	RDR Q/TIME	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 62: COMPUTE RDR Q/TIME(B)(4) EQ (RDR Q/TIME/HH * 3600) + (RDR Q/TIME/MM * 60) + RDR Q/TIME/SS At summary level, print with the (AV) suffix.
63	Writer Queue Time	WTR Q/TIME	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 63: COMPUTE WTR Q/TIME(B)(4) EQ (WTR Q/TIME/HH * 3600) + (WTR Q/TIME/MM * 60) + WTR Q/TIME/SS At summary level, print with (AV) suffix.
64	Turnaround Time	TAROUND/TIME	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 64: COMPUTE TAROUND/ TIME(B)(4) EQ (TAROUND/TIME/ HH * 3600) + (TAROUND/TIME/ MM * 60) + TAROUND/TIME/SS At summary level, print with (AV) suffix.
65	Elapsed Time	ELAPSED/TIME (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 65: COMPUTE ELAPSED/TIME EQ (ELAPSED/TIME/ HRS * 3600)
66	CPU Time	CPU/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 66: COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600)

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
67	Overhead Time	OVHEAD/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 67: COMPUTE OVHEAD/TIME EQ (OVHEAD/TIME/ HRS * 3600)
68	Wait Time	WAIT/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 68: COMPUTE WAIT/TIME EQ (WAIT/TIME/HRS * 3600)
69	Processor Time	PROCESS/TIME (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 69: COMPUTE PROCESS/TIME EQ (PROCESS/TIME/ HRS * 3600)
70	I/O Time	IO/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 70: COMPUTE IO/TIME EQ (IO/TIME/HRS * 3600)
71	Total Time	TOTAL/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 71: COMPUTE TOTAL/TIME EQ (PROCESS/TIME/ HRS + IO/TIME/HRS) * 3600
72	Storage Indicator	S/I	
73	Pages-in	PAGE IN/COUNT	
74	Pages-out	PAGE OUT/COUNT	
75	Total pages	TOTAL/PAGES	
76	Elapsed Paging Rate	ELAP/PAGE	At summary level, recalculated with: COMPUTE(T) ELAP/PAGE EQ TOTAL/PAGES / (ELAPSED/TIME/ HRS * 60)
77	CPU Paging Rate	CPU/PAGE	At summary level, recalculated with: COMPUTE(T) CPU/PAGE EQ TOTAL/PAGES / (CPU/TIME/HRS * 3600)

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
78	I/O Index	IO INDX	At summary level, recalculated with: COMPUTE(T) IO/INDX EQ TOTAL/IO COUNT / (CPU/TIME/HRS * 3600)
79	Credit Amount	not applicable	
80	Debit Amount	not applicable	
81	Budget Amount	not applicable	
82	Over/Under Budget	not applicable	
83	% Budget Spent	not applicable	
84	Group Code #1	G/1	
85	Group Code #2	G/2	
86	Group Code #3	G/3	
87	Partition-ID	P/N	
88	Reader Duration	RDR/DURATION (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 88: COMPUTE RDR/DURATION EQ (RDR/DURATION/ HRS * 3600)
89	Writer Duration	WTR/DURATION (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 89: COMPUTE WTR/DURATION EQ (WTR/DURATION/ HRS * 3600)
90	Elapsed Time-averaged	ELAPSED/TIME (AV)(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 90: COMPUTE ELAPSED/TIME EQ (ELAPSED/TIME/ HRS * 3600) Also, append the (AV) suffix to get the actual equivalent of data element 90.

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
91	CPU Time-averaged	CPU/TIME(AV) (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element 91: COMPUTE CPU/TIME EQ (CPU/TIME/HRS * 3600) Also, append the (AV) suffix to get the actual equivalent of data element 91.
92	# of Readers Used	NO/RD	
93	# of Printers Used	NO/PR	
94	# of Punches Used	NO/PU	
95	# of Tapes Used	NO/TP	
96	# Disks Used	NO/DK	
97	# Private Disks Used	NO/DU	
98	# Other Devices Used	NO/OT	
99	Blank Spaces	not applicable	
A0	Input Device Name	INPUT/DEVICE	
A1	Number of Swaps	SWAP/COUNT	
A2	Swap Pages-in	SWAP/PAGES IN	
A3	Swap Pages-out	SWAP/PAGES OUT	
A4	Total Swap Pages	TOTAL SWAP/PAGES	
A5	TPUT's Issued	TSO/TPUTS	
A6	TGET's Satisfied	TSO/TGETS	
A7	Active Time-in hours	ACTIVE/TIME/HRS	
A8	Active Time	ACTIVE/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element A8: COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/ HRS * 3600)

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
A9	Active Time-averaged	ACTIVE/TIME (AV)(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element A9: COMPUTE ACTIVE/TIME EQ (ACTIVE/TIME/ HRS * 3600) Also, append the (AV) suffix to get the actual equivalent of data element A9.
B0	Connect Time-in hours	CONNECT/TIME/ HRS	
B1	Connect Time	CONNECT/TIME (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element B1: COMPUTE CONNECT/TIME EQ (CONNECT/TIME/ HRS * 3600)
B2	Connect Time-averaged	CONNECT/TIME (AV)(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element B2: COMPUTE CONNECT/TIME EQ (CONNECT/TIME/ HRS * 3600) Also, append the (AV) suffix to get the actual equivalent of data element B2.
B3	Connect Charge	CONNECT/CHARGE	
B4	TSU Number	TSU/NBR	At summary level, use COUNT for number of records processed in the control group.
B5	Job Requeued Indicator	RQ/ID	
B6	Input Device Route Code	INP/RTE	
B7	Print Device Route Code	PRT/RTE	
B8	Allocation Time-in hours	ALLOC/TIME/HRS	
B9	Allocation Time	ALLOC/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element B9: COMPUTE ALLOC/TIME EQ (ALLOC/TIME/HRS * 3600)

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
C0	Allocation Time-averaged	ALLOC/TIME(AV) (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element C0: COMPUTE ALLOC/TIME EQ (ALLOC/TIME/HRS * 3600) Also, append the (AV) suffix to get the actual equivalent of data element C0.
C1	Resident Time-in hours	RES/TIME/HRS	
C2	Resident Time	RES/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element C2: COMPUTE RES/TIME EQ (RES/TIME/HRS * 3600)
C3	Resident Time-averaged	RES/TIME(AV) (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element C3: COMPUTE RES/TIME EQ
C4	CPU Time (SRB)-in hours	SRB/TIME/HRS	
C5	CPU Time (SRB)	SRB/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element C5: COMPUTE SRB/TIME EQ (SRB/TIME/HRS * 3600)
C6	CPU Time (SRB)-averaged	SRB/TIME(AV) (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element C6: COMPUTE SRB/TIME EQ (SRB/TIME/HRS * 3600) Also, append the (AV) suffix to get the actual equivalent of data element C6.
C7	CPU Time (TCB)-in hours	TCB/TIME/HRS	
C8	CPU Time (TCB)	TCB/TIME(HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element C8: COMPUTE TCB/TIME EQ (TCB/TIME/HRS * 3600)

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
C9	CPU Time (TCB)-averaged	TCB/TIME(AV) (HS)	Use the following statement in your CA JARS Wizard run to get the actual equivalent of data element C9: COMPUTE TCB/TIME EQ (TCB/TIME/HRS * 3600) Also, append the (AV) suffix to get the actual equivalent of data element C9.
D0	CPU Index	CPU/IDX	At summary level, recalculated with: COMPUTE(T) CPU/IDX EQ (SRB TIME/HRS * 100) / CPU/TIME/HRS
D1	Service Units	SERVICE/UNITS	
D2	Service Rate	SERV/RATE	At summary level, recalculated with: COMPUTE(T) SERV/RATE EQ SERVICE/UNITS / (ACTIVE/TIME/ HRS * 3600)
D3	Page Seconds	PAGE/SECONDS	
D4	Performance Group Number	PER/GRP	
D5	Day of week indicator	D/I	
D6	Day of week	not applicable	
D7	Group Code #4	G/4	
D8	Group Code #5	G/5	
D9	Group Code #6	G/6	
E0	Forms Count	FORMS/COUNT	
E1	SYSOUT Class	O/C	
E2	SYSOUT Type Code	O/T	
E3	SYSOUT Intervention Code	I/C	
E4	Output Device Name	OUTPUT/DVC NAME	
E5	Lines Printed (Standard)	STD LINES/PRNTD	
E6	Lines Printed (Special)	SPCL LINES/PRNTD	
E7	Pages Printed (Standard)	STD PAGES/PRNTD	
E8	Pages Printed (Special)	SPCL PAGES/PRNTD	
E9	Pages Printed (Total)	TOT PAGES/PRNTD	

DE ID	Data Element Name	CAWGJR70 Glossary Item	Notes
F0	# Tape Mounts	TAPE/MOUNTS	
F1	# Disk Mounts	DISK/MOUNTS	
F2	2nd Accounting Field	USER INFO2	
F3	3rd Accounting Field	USER INFO3	
F4	User Character Field	USER/FIELD	
F5	User Count Field	USER/COUNT	
F6	User Time Field	USER/TIME	
F7	Control Break Count	not applicable	
F8	Absorption Rate	ABSORP/RATE	At summary level, recalculated with: COMPUTE(T) ABSORP/RATE EQ SERVICE/UNITS / (RES/TIME/HRS * 3600)
F9	SMF Audit Flags	AUDIT/FLAGS	
G0	I/O Count by Device	not applicable	
G1	RJE Connect Time	CNT/TIME/RJE/ HRS	
G2	RJE Transmission Count	RJE LINE/EXCP CT	
G3	RJE Line Errors	RJE LINE/ERRORS	
G4	RJE Not Acknowledged	RJE LINE/NAK CT	
G5	RJE Time-Out Count	RJE LINE/TME OUT	
G6	RJE Data Checks	RJE LINE/DATA CK	
G7	RJE Active Time	ACT/TIME/RJE/ HRS	
G8	RJE Invalid Logon Count	RJE/INV LG	
G9	RJE Line Password	RJE/LINE/ PASSWRD	
H0	RJE Line Remote Name	REMOTE/NAME	
H1	RJE Line Name	RJE LINE/NAME	
H2	RJE Charge	RJE/CHARGE	
H3	RJE Session Count	not applicable	

Appendix E: CAWGJRD2 Glossary

Name	Pos	Len	Format	Suffixes	Description
CAWGJRD2/RECORD	1	374	H	NV	RECORD IN HEX
USER/ID	1	8	C		AUTHORIZATION ID
SYS/ID	9	4	C		SMF SYSTEM ID
FILLER01	13	1	B		FILLER
CONNECT/DATE	14	3	P	YYDDD NV	DATE CONNECTED
CONNECT/TOD	17	4	B	.2 HS NV	TIME CONNECTED, 1/100 SEC
CONNECTION/NAME	21	8	C		USERS HOST
PLAN/NAME	29	8	C		PLAN NAME
CORRELATION/ID	37	12	C		CORRELATION ID
NETWORK/ID	49	16	C		NETWORK ID
SUBSYS/ID	65	4	C		WHAT YOU NAMED DB2
RELEASE/IND	69	1	B	NV	DB2 RELEASE INDICATOR
FILLER02	70	1	B	NV	RESERVED FIELD
REASON/CODE	71	2	B	NV	, FOR ACCOUNTING INVOCATION
IFCID#	73	4	B	NV	, SEQUENCE NUMBER FOR
DESTINATION#	77	4	B	NV	, SEQUENCE NUMBER FOR
TRACE#/MASK	81	4	B	NV	ACTIVE TRACE NUMBER MASK
COMMITS	85	4	B		NUMBER OF COMMITS
ABORTS	89	4	B		NUMBER OF ABORTS
CONNECT/TIME	93	4	B	.2 KS	ELAPSED 1/100 SEC
CPU/TIME	97	4	B	.2 KS	TCB CPU 1/100 SEC
SRB/TIME	101	4	B	.2 KS	SRB CPU 1/100 SEC
IO/WAIT/TIME	105	4	B	.2 KS	IO WAIT 1/100 SEC (CLASS 3)
LOCK/TIME	109	4	B	.2 KS	LL TIME 1/100 SEC (CLASS 3)

Name	Pos	Len	Format	Suffixes	Description
WAIT/EVENTS	113	4	B		WAIT TRACE EVENTS (CLASS 3)
TRACE/EVENTS	117	4	B		ENTRY/EXIT EVENTS (CLASS 2)
GET/PAGE/BP0	121	4	B		NUMBER OF GET PAGE REQUEST
EXPAND/BP0	125	4	B		" CAUSING BP EXPANSION
SYSPAGE/UP/BP0	129	4	B		UPDATE SYSPAGE
UWPAGE/UP/BP0	133	4	B		UPDATE UNIT OF WORK PAGE
READ/SYNC/BP0	137	4	B		SYNCHRONOUS READ REQUEST
READ/PREF/BP0	141	4	B		REPFETCH READ REQUEST
GET/PAGE/BP1	145	4	B		NUMBER OF GET PAGE REQUEST
EXPAND/BP1	149	4	B		" CAUSING BP EXPANSION
SYSPAGE/UP/BP1	153	4	B		UPDATE SYSPAGE
UWPAGE/UP/BP1	157	4	B		UPDATE UNIT OF WORK PAGE
READ/SYNC/BP1	161	4	B		SYNCHRONOUS READ REQUEST
READ/PREF/BP1	165	4	B		REPFETCH READ REQUEST
GET/PAGE/BP2	169	4	B		NUMBER OF GET PAGE REQUEST
EXPAND/BP2	173	4	B		" CAUSING BP EXPANSION
SYSPAGE/UP/BP2	177	4	B		UPDATE SYSPAGE
UWPAGE/UP/BP2	181	4	B		UPDATE UNIT OF WORK PAGE
READ/SYNC/BP2	185	4	B		SYNCHRONOUS READ REQUEST
READ/PREF/BP2	189	4	B		REPFETCH READ REQUEST
GET/PAGE/BP32	193	4	B		NUMBER OF GET PAGE REQUEST
EXPAND/BP32	197	4	B		" CAUSING BP EXPANSION

Name	Pos	Len	Format	Suffixes	Description
SYSPAGE/UP/BP32	201	4	B		UPDATE SYSPAGE
UWPAGE/UP/BP32	205	4	B		UPDATE UNIT OF WORK PAGE
READ/SYNC/BP32	209	4	B		SYNCHRONOUS READ REQUEST
READ/PREF/BP32	213	4	B		REPFETCH READ REQUEST
DEADLOCK	217	4	B		NUMBER OF DEADLOCKS
SUSPEND/LOCK	221	4	B		HAD TO WAIT FOR LOCK, COUNT
TIMEOUT/LOCK	225	4	B		TIMED-OUT WAITING, COUNT
ESCALATE/SHR	229	4	B		LOCK ESCALATED TO SHARE MODE
ESCALATE/EXC	233	4	B		LOCK ESCALATED TO EXCL. MODE
LOCKS/HELD	237	4	B		MAXIMUM PAGE LOCKS HELD
SELECT	241	4	B		SQL SELECT
INSERT	245	4	B		SQL INSERT
UPDATE	249	4	B		SQL UPDATE
DELETE	253	4	B		SQL DELETE
DESCRIBE	257	4	B		SQL DESCRIBE
PREPARE	261	4	B		SQL PREPARE
OPEN	265	4	B		SQL OPEN
CLOSE	269	4	B		SQL CLOSE
CREATE/TABLE	273	4	B		SQL CREATE TABLE
CREATE/INDEX	277	4	B		SQL CREATE INDEX
CREATE/TABLESP	281	4	B		SQL CREATE TABLESPACE
CREATE/SYNONYM	285	4	B		SQL CREATE SYNONYM
CREATE/DATABASE	289	4	B		SQL CREATE DATABASE
CREATE/STOGROUP	293	4	B		SQL CREATE STORAGE GROUP
CREATE/VIEW	297	4	B		SQL CREATE VIEW

Name	Pos	Len	Format	Suffixes	Description
DROP/INDEX	301	4	B		SQL DROP INDEX
DROP/TABLE	305	4	B		SQL DROP TABLE
DROP/TABLESP	309	4	B		SQL DROP TABLESPACE
DROP/DATABASE	313	4	B		SQL DROP DATABASE
DROP/SYNONYM	317	4	B		SQL DROP SYNONYM
DROP/STOGROUP	321	4	B		SQL DROP STOGROUP
DROP/VIEW	325	4	B		SQL DROP VIEW
ALT/STOGROUP	329	4	B		SQL ALTER STOGROUP
FETCH	333	4	B		SQL FETCH
ALT/TABLESP	337	4	B		SQL ALTER TABLESPACE
ALT/TABLE	341	4	B		SQL ALTER TABLE
ALT/INDEX	345	4	B		SQL ALTER INDEX
COMMENT/ON	349	4	B		SQL COMMENT ON
LOCK/TABLE	353	4	B		SQL LOCK TABLE
GRANT	357	4	B		SQL GRANT
REVOKE	361	4	B		SQL REVOKE
BIND	365	4	B		SQL INCREMENTAL BIND
LABEL/ON	369	4	B		SQL LABEL ON

Appendix F: CAWGJRUV Glossary

Name	Pos	Len	Format	Suffixes	Description
CAWGJRUV/RECORD	1	112	H		
USER/NAME	1	8	C		USERS NAME
USER/CODE	9	2	C		USERS 2 DIGIT USERCODE
USER/TYPE	11	1	C		1 = DBA, 0 = NON-DBA
OK/EDITOR	12	1	C		PERMISSION TO USE EDITOR
OK/UGEN	13	1	C		PERMISSION TO USE UGEN
OK/UPREP	14	1	C		PERMISSION TO USE UPREP
OK/DATADEF	15	1	C		PERMISSION TO USE DATA DEF
OK/LUW	16	1	C		PERMISSION TO USE LUW
OK/FUNCTIONS	17	1	C		PERMISSION TO USE FUNCTIONS
OK/OVERCPU	18	1	C		PERM N TO OVERRIDE CPU/IO
SP1	19	2	C		
LIMIT/CPU	21	4	B		LMT ON CPU SECS BEFORE WARN
LIMIT/IO	25	4	B		LMT OF I/O
USED/CPU	29	4	B		CPU USED SINCE LAST CLEAR
USED/IO/REQ	33	4	B		I/O REQUESTS SINCE LAST CLR
USED/ACTUAL/IO	37	4	B		ACTUAL I/O SINCE LAST CLEAR
CPU/WARNINGS	41	4	B		COUNT CPU WARNINGS GIVEN
IO/WARNINGS	45	4	B		COUNT I/O WARNINGS GIVEN
RESOURCE/WAIT	49	4	B		USR WAITED FOR RESOURCE CNT
DATADEF/CMDS	53	4	B		DATA DEFINITION COMMANDS
DATADICT/UPDATES	57	4	B		DATA DICT UPDATES SINCE CLR
INTERGRITY/VIOL	61	4	B		INTEGRITY VIOLATIONS
USER/ROLLED/LUW	65	4	B		USER ROLLED BACK LUW
USER/DEADLOCK	69	4	B		USER CAUSED A DEADLOCK
FORCED	73	4	B		TIMES FORCED SINCE CLEAR
CPU/SINCE/THOLD	77	4	B		CPU SINCE LAST THRESHOLD

Name	Pos	Len	Format	Suffixes	Description
IO/SINCE/THOLD	81	4	B		I/O SINCE LAST THRESHOLD
USERID	85	8	C		USERID OF MACHINE
INTERFACE/FLGS	93	4	B		INTERFACE FLAGS
TIME/CONNECT	97	8	C		TIME USER CONNECTED TO DB
TIME/DISCONNECT	105	8	C		TIME USER DISCONNECTED FR DB

Appendix G: CAWGDX60 Glossary

The CAWGDX60 glossary documented here is used for the CA JARS DSA Option CA JARS Wizard reporting. The WIZPG program can be used to create a detailed listing of the CAWGDX60 glossary layout.

Name	Pos	Len	Format	Suffixes
CAWGDX60/RECORD	1	511	C	
HEX/RECORD	1	511	H	
CAT 1/COUNT	1	4	B	
CAT 2/COUNT	5	4	B	
CAT 3/COUNT	9	4	B	
CAT 1/TRKA	13	4	B	
CAT 2/TRKA	17	4	B	
CAT 3/TRKA	21	4	B	
CAT 1/TRKU	25	4	B	
CAT 2/TRKU	29	4	B	
CAT 3/TRKU	33	4	B	
CAT 1/AMOUNT	37	8	P	.2 L9
CAT 2/AMOUNT	45	8	P	.2 L9
CAT 3/AMOUNT	53	8	P	.2 L9
ACCOUNT/AMOUNT	61	8	P	.2 L9
TRACKS/ALLOCATED	73	8	B	
TRACKS/USED	81	8	B	
ACCOUNT/NUMBER	89	16	C	
VTOC SCAN/F DT T	6	24	H	
VTOC SCAN/BASEDA	6	4	B	
VTOC SCAN/DT TM	10	20	C	
VTOC SCAN/DATE	10	11	C	
VTOC SCAN/TIME	22	8	C	
BILL STRT/F DT T	30	24	H	

Name	Pos	Len	Format	Suffixes
BILL STRT/BASEDA	30	4	B	
BILL STRT/DT TM	34	20	C	
BILL STRT/DATE	34	11	C	
BILL STRT/TIME	46	8	C	
BILL END/F DT T	54	24	H	
BILL END/BASEDA	54	4	B	
BILL END/DT TM	58	20	C	
BILL END/DATE	58	11	C	
BILL END/TIME	70	8	C	
UPDATE/FULL DT T	78	24	H	
UPDATE/BASEDATE	78	4	B	
UPDATE/DATE TIME	82	20	C	
UPDATE/DATE	82	11	C	
UPDATE/TIME	94	8	C	
TOTAL/SORT KEY	5	52	C	
REC/ID	5	1	C	
REST/SORT KEY	6	50	C	
DATASET/NAME	6	44	C	
VOLSER	50	6	C	
V F	56	1	C	
FIRST/QUALIFIER	57	8	C	
CR/DATE	65	4	P	
EXPIR/DATE	69	4	P	
NUM/EXT	73	1	B	
L R/DATE	74	4	P	
L R/FLG	78	1	C	
DSORG	79	2	H	
RECFM	81	1	H	
BLKSIZE	82	2	B	
LRECL	84	2	B	

Name	Pos	Len	Format	Suffixes
OTHER/VTOC	86	1	H	
DEVICE/ATTRIBUTE	87	14	H	
UNIT/TYPE	101	1	H	
TRKS/ALLOC	102	4	B	
TRKS/USED	106	4	B	
SMS/FLAG	110	1	C	
STORAGE/CLASS	111	8	C	
MGMT/CLASS	119	8	C	
DATA/CLASS	127	8	C	
POSIX/FLAG	135	1	C	
DT TM/ALLOC	136	8	H	
TIME/ALLOC	136	4	B	
DATE/ALLOC	140	4	P	
CURR/CHARGE	144	8	P	.2
RATE	152	8	P	.5
RATE/TYPE	160	1	H	
MOD/FLAG	161	1	C	
SMS/FACTOR	162	2	P	
RESERVED/EXPAND	164	14	H	
SYS/ID	178	4	H	
DT TM/SCR	182	8	H	
TIME/SCR	182	4	B	
DATE/SCR	186	4	P	
INT 1/STAT	190	1	H	
INTERVAL 1	191	65	H	
INT 1/START DT T	191	16	H	
INT 1/START BASE	191	4	B	
INT 1/START DT	195	6	C	
INT 1/START TM	201	6	C	
INT 1/END DT TM	207	16	H	

Name	Pos	Len	Format	Suffixes
INT 1/END BASE	207	4	B	
INT 1/END DT	211	6	C	
INT 1/END TM	217	6	C	
INT 1/DURATION	223	4	B	
INT 1/TOT TRACKS	227	4	B	
INT 1/TRACK SECS	231	4	B	
INT 1/TRACK HRS	239	8	P	.5
INT 1/KBTYE HRS	247	8	P	
INT 1/FUNC	263	1	C	
INT 2/STAT	264	1	H	
INTERVAL 2	265	65	H	
INT 2/START DT T	265	16	H	
INT 2/START BASE	265	4	B	
INT 2/START DT	269	6	C	
INT 2/START TM	275	6	C	
INT 2/END DT TM	281	16	H	
INT 2/END BASE	281	4	B	
INT 2/END DT	285	6	C	
INT 2/END TM	291	6	C	
INT 2/DURATION	297	4	B	
INT 2/TOT TRACKS	301	4	B	
INT 2/TRACK SECS	305	4	B	
INT 2/TRACK HRS	313	8	P	.5
INT 2/KBYTE HRS	321	8	P	
INT 2/FUNC	337	1	C	
JOBNAME	338	8	C	
STEPNAME	346	8	C	
JOB/CTR	346	4	B	
STEP/CTR	350	4	B	
JOB/TYPE	354	1	C	

Name	Pos	Len	Format	Suffixes
COMP/ACCT NUM	355	16	C	
ACCT NO/LEN	371	1	B	
FULL/ACCT NUM	372	140	C	

Name	Pos	Len	Format	Suffixes	Description
JR70CBF/RECORD	1	8740	C		RECORD IN CHARACTER FORMAT
HEX RECORD	1	8740	H		RECORD IN HEXADECIMAL FORMAT
RCD DESCRIPTOR	1	2	B	NV	RECORD LENGTH FROM RDW
RDW	1	2	B	NV	RECORD LENGTH FROM RDW
CPU/ID	5	4	C		ID OF CPU WHERE JOB WAS EXECUTED
SMF/ID	5	4	C		SMF ID
JOB/TIME/STAMP	9	7	H		UNIQUE JOB DATE AND TIME STAMP
INTERVAL/START	9	4	P	HM NV	INTERVAL CHOPPER START TIME
INTERVAL/LENGTH	13	3	P	HM NV	INTERVAL CHOPPER INTERVAL LENGTH
RDR/DATE	9	3	P	YYDDD	READER START DATE FOR THE JOB
SES/START/DATE	9	3	P	YYDDD	SESSION START DATE FOR NETWORK USER
RDR/START/TIME	12	8	P	.4 NV HM	
SES/START/TIME	12	8	P	.4 NV HM	SESSION START TIME FOR NETWORK USER
STEP/NO	20	2	B	NT L4	RELATIVE STEP NUMBER WITHIN THE JOB
STEP/NBR	20	2	B	NT L4	NUMBER OF STEPS AT JOB SUMMARY LEVELS
SHIFT	22	1	C		SHIFT IN WHICH THE JOB WAS RUN
GREG/DATE	23	6	C	YYMMDD	STEP START DATE
DATE	23	6	C	YYMMDD	STEP START DATE
YEAR	23	2	C		STEP START YEAR
MONTH	25	2	C		STEP START MONTH
DAY	27	2	C		STEP START DAY
JOB NAME	29	8	C		JOB NAME FROM THE JOB STATEMENT
JOBNAME	29	8	C		JOB NAME FROM THE JOB STATEMENT

Name	Pos	Len	Format	Suffixes	Description
USER ID	29	8	C		VM USER IDENTITY
TERMINAL/ID	29	8	C		TERMINAL ID
START/DATE	37	3	P	YYDDD	DATE INITIATOR/ USER SELECTED THIS STEP
LOGON DATE	37	3	P	YYDDD	VM LOGON DATE
SYSOUT/DATE	37	3	P	YYDDD	SYSOUT START DATE- Y RECORD
START/TIME	40	8	P	.4 NV HM	
CONV/START/TIME	40	8	P	.4 NV HM	APPC CONVERSATION START TIME
TRAN/START/TIME	40	8	P	.4 NV HM	APPC TRANSACTION START TIME
LOGON TIME	40	8	P	.4 NV HM	LOGON TIME
SYSOUT/TIME	40	8	P	.4 NV HM	SYSOUT START TIME
STOP/DATE	48	3	P	YYDDD	DATE STEP TERMINATED
LOGOFF DATE	48	3	P	YYDDD	LOGOFF DATE
STOP/TIME	51	8	P	.4 NV HM	TIME STEP TERMINATED
CONV/STOP/TIME	51	8	P	.4 NV HM	APPC CONVERSATION STOP TIME
TRAN/STOP/TIME	51	8	P	.4 NV HM	APPC TRANSACTION STOP TIME
LOGOFF TIME	51	8	P	.4 NV HM	LOGOFF TIME
ELAPS/TIME	59	8	P	.4 HM SP	TOTAL ELAPSED TIME WITHIN THE SHIFT
ELAPSED/TIME	59	8	P	.4 HM SP	TOTAL ELAPSED TIME WITHIN THE SHIFT
CONNECT/TIME	59	8	P	.4 HM SP	CONNECT TIME
ADJST/TIME	67	8	P	.4 NV HM	START TIME ADJUSTED W/IN THE SHIFT
ADJSP/TIME	75	8	P	.4 NV HM	STOP TIME ADJUSTED W/IN THE SHIFT
REC/TYPE	83	1	C		CA JASPER/JA RECORD TYPE
SUB/REC/TYPE	84	1	C		CA JASPER/JA SUB-RECORD TYPE
SYS/TYPE	85	3	C		CA JASPER GENERATED - DOS
CL	85	3	C		CLASS z/OS
JOB/START/TIME	88	8	P	.4 HM NV	

Name	Pos	Len	Format	Suffixes	Description
NET/START/TIME	88	8	P	.4 HM NV	NETWORK START TIME FOR POST-MIDNIGHT
PHYS/CPUID	96	1	C		ID OF CPU WHERE EVENT OCCURED
REL/LEVEL	97	3	C		RELEASE AND MODIFICATION LEVEL
REL/VER	97	1	C		RELEASE VERSION
REL/REL	98	1	C		RELEASE LEVEL
REL/MOD	99	1	C		MODIFICATION LEVEL
SAVED/REC/IND	100	1	C		SAVED RECORD INDICATOR
SORT/RUN/TYPE	101	1	C		
VM/TYPE	102	1	C		
DAY/OF/WK	103	1	C		DAY OF WEEK INDICATOR
RESERVEDB	104	4	B	NT	RESERVED COMMON SECTION BINARY FIELD
VOL/RENT/TIME	104	4	B	NT	TAPE VOLUME RENT TIME
SPLIT/IND	108	1	C		SPLIT RECORD INDICATOR
RESERVEDC	109	7	C		RESERVED COMMON SECTION CHARACTER FIELD
DATA/LENGTH	116	2	B	NT	
A..A RECORD	1	1122	H		ADVANCED FUNCTION PRINTING RECORD
A.PWVS DESC	118	16	C		POWER DESCRIPTION
A.PWVS/ DESCRIPTION	118	16	C		POWER DESCRIPTION
A.PWVS/JOBNAME	134	8	C		POWER JOB NAME
A.AA/PWVS/ JOBNAME	134	8	C		FORCE DOS FORMATTING OF OUTPUT RECORD
A.REPORT NAME	134	8	C		NAME OF REPORT
A.PWVS/SEQNO	142	2	B	NT	POWER SEQUENCE NUMBER
A.PWVS/CANCEL	144	4	C		POWER CANCEL CODE
A.XPCC/APPL ID	148	8	C		XPCC APPLICATION IDENTIFIER
A.DIV/CODE	156	4	C		DIVISION CODE
A.CHARGE/CODE	160	4	C		USER DEFINED CHARGE CODE

Name	Pos	Len	Format	Suffixes	Description
A.CHG/CODE	160	4	C		USER DEFINED CHARGE CODE
A.APPL/CODE	164	8	C		USER DEFINED APPLICATION CODE
A.PROG/CODE	172	4	C		USER DEFINED PROGRAM CODE
A.PGMR/CODE	172	4	C		USER DEFINED PROGRAM CODE
A.XTRA BYTE/COUNT	176	2	B		NUMBER OF BYTES OF ADDITIONAL DATA
A.LAST/ENTRY	176	2	B		LAST ENTRY IN A. RECORD
A.LASTENT/ RESERVED	176	2	B		LAST ENTRY IN A. RECORD
A.XTRA BYTES	178	945	H	DOS	ADDITIONAL AFP DATA
D..D RECORD	1	133	H		DEVICE ACTIVITY RECORD
D.DEV/ADDR	118	3	C		DEVICE ADDRESS - CUU
D.DEV/CL	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
D.DEV/CLASS	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
D.DEV/TYPE	122	4	C		DEVICE TYPE E.G. 2400, 3350 ETC.
D.SIO	126	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
D.REQUESTS/SERV	126	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
D.SHIFT/LENGTH	130	4	B	.4 HM SP NT	
E..E RECORD	1	8740	H		STEP END RECORD
E.STEP/COMP/CODE	118	4	C		STEP COMPLETION CODE
E.STEP/TERM/CODE	122	1	C		STEP TERMINATION CODE
E.PROGRAM/NAME	123	8	C		PROGRAM NAME FROM EXEC STATEMENT
E.PART	131	2	C		PARTITION ID
E.DISK/DATA/SETS	133	2	B		NUMBER OF DISK DRIVES USED IN THE STEP
E.NO OF/DISKS	133	2	B		NUMBER OF DISK DRIVES USED IN THE STEP
E.NO OF/TAPES	135	2	B		NUMBER OF TAPE DRIVES USED IN THE STEP
E.TAPE/DATA/SETS	135	2	B		NUMBER OF TAPE DRIVES USED IN THE STEP

Name	Pos	Len	Format	Suffixes	Description
E.NO OF/OTHER	137	2	B		NUMBER OF OTHER DEVICES USED IN STEP
E.OTHER/DATA/SETS	137	2	B		NUMBER OF OTHER DEVICES USED IN STEP
E.NO OF/MISC	139	2	B		NUMBER OF MISCELLANEOUS DEVICES USED
E.MISC/DATA/SETS	139	2	B		NUMBER OF NON DISK/TAPE DEVICES USED
E.DISK/SIO	141	4	B	L7 SP	NUMBER OF DISK SIO ISSUED IN THE STEP
E.DISK/EXCP	141	4	B	L7 SP	NUMBER OF DISK SIO ISSUED IN THE STEP
E.TAPE/SIO	145	4	B	L7 SP	NUMBER OF TAPE SIO ISSUED IN THE STEP
E.TAPE/EXCP	145	4	B	L7 SP	NUMBER OF TAPE SIO ISSUED IN THE STEP
E.READER/SIO	149	4	B	L7 SP	NUMBER OF RDR SIO ISSUED IN THE STEP
E.READER/EXCP	149	4	B	L7 SP	NUMBER OF RDR SIO ISSUED IN THE STEP
E.PRINTER/SIO	153	4	B	L7 SP	NUMBER OF PRT SIO ISSUED IN THE STEP
E.PRINTER/EXCP	153	4	B	L7 SP	NUMBER OF PRT SIO ISSUED IN THE STEP
E.PUNCH/SIO	157	4	B	L7 SP	NUMBER OF PUN SIO ISSUED IN THE STEP
E.PUNCH/EXCP	157	4	B	L7 SP	NUMBER OF PUN SIO ISSUED IN THE STEP
E.OTHER/SIO	161	4	B	L7 SP	NUMBER OF OTHER SIO ISSUED IN STEP
E.OTHER/EXCP	161	4	B	L7 SP	NUMBER OF OTHER SIO ISSUED IN STEP
E.MISC/SIO	165	4	B	L7 SP	NUMBER OF NON DISK/TAPE SIO ISSUED
E.MISC/EXCP	165	4	B	L7 SP	NUMBER OF NON DISK/TAPE SIO ISSUED

Name	Pos	Len	Format	Suffixes	Description
E.CPU/TIME	169	8	P	.4 MS SP	STEP CPU TIME
E.PAGE/INS	177	4	B	L7 SP	NUMBER OF PAGE-INS DURING EXECUTION
E.PAGE/OUTS	181	4	B	L7 SP	NUMBER OF PAGE-OUTS DURING EXECUTION
E.ACTIVE/TIME	185	8	P	.4 HM SP	STEP ACTIVE TIME
E.JOB/CL	193	1	C		POWER JOB CLASS
E.JOB/CLASS	193	1	C		POWER JOB CLASS
E.JOB/PRTY	194	2	B	NT	POWER JOB PRIORITY
E.JOB/FROM/USER	196	8	C		JOB FROM USER
E.RJE/IN	196	3	C		INPUT REMOTE DEVICE IN DECIMAL
E.RJE/OUT	204	3	C		OUTPUT REMOTE DEVICE IN DECIMAL
E.JOB/TO/USER	204	8	C		JOB TO USER
E.RUN/TYPE	212	1	C		RUN TYPE - P,T,C,N,R OR U
E.SUB/RUN/TYPE	213	1	C		SUB RUN TYPE
E.DOS/JOBNAME	214	8	C		DOS JOB NAME
E.DOS/DESC	222	16	C		DOS DESCRIPTION
E.DOS/ DESCRIPTION	222	16	C		DOS DESCRIPTION
E.PWVS DESC	238	16	C		POWER DESCRIPTION
E.PWVS/ DESCRIPTION	238	16	C		POWER DESCRIPTION
E.PWVS/ JOBNAME	254	8	C		POWER JOB NAME
E.AA/PWVS/ JOBNAME	254	8	C		FORCE DOS FORMATTING OF OUTPUT RECORD
E.PWVS/SEQNO	262	2	B	NT	POWER SEQUENCE NUMBER
E.PWVS/ABEND	264	1	C		POWER JOB ABEND INDICATOR
E.PWVS/CANCEL	265	4	C		POWER CANCEL CODE
E.STEP/FLAG	269	1	C		STEP INDICATOR S OR L
E.RESERVED/NOJ	270	1	N	SP	RESERVED NUMBER OF VSE JOBS
E.DIV/CODE	271	4	C		DIVISION CODE
E.CHARGE/CODE	275	4	C		USER DEFINED CHARGE CODE

Name	Pos	Len	Format	Suffixes	Description
E.CHG/CODE	275	4	C		USER DEFINED CHARGE CODE
E.APPL/CODE	279	8	C		USER DEFINED APPLICATION CODE
E.PROG/CODE	287	4	C		USER DEFINED PROGRAM CODE
E.PGMR/CODE	287	4	C		USER DEFINED PROGRAM CODE
E.SPOOL/LINES	291	4	B	SP	NUMBER OF SPOOLED PRINT LINES FOR STEP
E.SPOOL/CARDS	295	4	B	SP	NUMBER OF SPOOLED PUNCH CARDS FOR STEP
E.SPOOL/PAGES	299	4	B	SP	NUMBER OF SPOOLED PAGES FOR STEP
E.OVERHEAD	303	8	P	.4 MS SP	STEP OVERHEAD TIME
E.ALLBOUND	311	8	P	.4 MS SP	STEP ALLBOUND TIME
E.RESERVED/NOP	319	1	N		RESERVED NUMBER OF POWER JOBS
E.RESERVED/NOS	320	1	N		RESERVED NUMBER OF EXECUTION STEPS
E.HIGH/CORE	321	3	P	NT	CORE USED
E.SETUP/ITIME	324	8	P	.4 HS	SETUP-IDLE TIME IN SECONDS
E.RESERVED PACK1JD	332	8	P		RESERVED PACKED FIELD - JOB START DATE
E.RESERVED PACK2	340	8	P		RESERVED PACK FIELD 2
E.RESERVED PACK3	348	8	P		RESERVED PACKED FIELD 3
E.CORE/IND	356	1	C		STORAGE INDICATOR
E.TOTAL/PAGES	357	5	P	SP	TOTAL PAGE COUNT
E.TOTAL/COUNT	362	5	P	SP	TOTAL I/O COUNT
E.NO/RDR	367	3	P		NO. OF READERS USED
E.NO/PTR	370	3	P		NO. OF PRINTERS USED
E.NO/PCH	373	3	P		NO. OF PUNCHES USED
E.NO/DKP	376	3	P		NO. OF DISK-PRIVATE USED
E.NO/DEVICES	379	3	P		NO. OF DEVICES USED
E.NO/SIO/ENT	382	3	P		NO. SIO ENTRIES ON E REC
E.NODEVIC/ RESERVED	382	3	P		NO. OF DEVICES USED

Name	Pos	Len	Format	Suffixes	Description
E.USER/FIELD1	385	16	C		USER DATA FIELD
E.USER/FIELD2	401	5	P	SP	USER COUNT FIELD
E.USER/FIELD3	406	8	P	SP	USER TIME FIELD
E.USER/FIELD4	414	4	B		USER FIELD 4 - BINARY
E.USER/FIELD5	418	4	B		USER FIELD 5 - BINARY
E.USER/FIELD6	422	4	B		USER FIELD 6 - BINARY
E.USER/FIELD7	426	4	B		USER FIELD 7 - BINARY
E.ORIG/SEQNO	430	2	B	NT	ORIGINAL SEQUENCE NUMBER
E.EXEC/NODE	432	8	C		NAME OF EXECUTION NODE
E.FROM/NODE	440	8	C		NAME OF FROM NODE
E.FROM/USER	448	8	C		ID OF ORIGINATING USER
E.UGPSW/ RESERVED	456	1	C		WIZUGP SIO CHOPPER PROCESSED
E.RESERVEDER	457	8	C		RESERVED CHAR FIELD
E.DEACT	465	4	B	.4 MS SP	STEP DEACTIVATION TIME
E.WORK/SET	469	4	B	NT	SIZE OF STEP WORKING SET
E.OPER/ID	473	2	C		OPERATOR ID
E.CKHPT/FLAG	475	1	C		CHECKPOINT INDICATOR
E.LEAVE/FLAG	476	1	C		
E.NO OF/JOBS	477	4	B	SP	= 1 FOR 1ST STEP, =0 FOR ALL OTHERS
E.NO OF/DOS JOBS	477	4	B	SP	NUMBER OF DOS JOBS WITHIN PVS JOB
E.NO OF/PWVS JOBS	481	4	B	SP	NUMBER OF POWER JOBS
E.NO OF/STEPS	485	4	B	SP	NUMBER OF EXECUTION STEPS
E.RESERVEDC	489	8	C		RESERVED CHARACTER BYTES
E.RESERVED/CPU	497	8	P	MS SP	CPU TIME IN WHOLE SECONDS
E.RESERVED/OVHD	505	8	P	MS SP	OVHD TIME IN WHOLE SECONDS
E.RESERVED/WAIT	513	8	P	MS SP	WAIT TIME IN WHOLE SECONDS
E.NETWORK/ACCT NO	521	8	C		NETWORK ACCOUNT NUMBER
E.RESERVED/ POWEXT	529	20	C		LAST 20 BYTES OF POWER EXTENTION

Name	Pos	Len	Format	Suffixes	Description
E.LAST/ENTRY	529	20	C		LAST ENTRY IN E. RECORD FOR WIZUGP
E.LASTENT/ RESERVED	529	20	C		LAST ENTRY IN E. RECORD FOR WIZUGP
E.SIO/ENTRIES	549	8192	H	DOS	SIO ENTRIES FOR ALL DEVICES
F..F RECORD	1	224	H		ICCF ACCOUNT RECORD
F.LIBRARY	118	2	B	NT	LIBRARY ASSIGNED TO THE USER
F.NO OF/LOGON	120	4	B	SP	NUMBER OF LOGONS
F.ENTER/REQS	124	4	B	SP	NUMBER OF ENTER KEY REQUESTS
F.EXEC/REQS	128	4	B	SP	NUMBER OF EXECUTION REQUESTS
F.EXEC/UNITS	132	8	P	.4 KH SP	NUMBER OF EXECUTION UNITS USED
F.SPACE/USAGE	140	4	B	SP	TOTAL SPACE USAGE
F.ICCF/TOTAL	144	8	P	.4 KH SP	
F.INTERACT/TIME	152	8	P	.4 KH SP	
F.RESERVED/LTIME	160	8	P	MS SP	LOGON TIME IN WHOLE SECONDS
F.RESERVED/PTIME	168	8	P	MS SP	PART TIME IN WHOLE SECONDS
F.USER/FIELD1	176	16	C		USER DATA FIELD
F.USER/FIELD2	192	5	P	SP	USER COUNT FIELD
F.USER/FIELD3	197	8	P	SP	USER TIME FIELD
F.DIV/CODE	205	4	C		DIVISION CODE
F.CHARGE/CODE	209	4	C		USER DEFINED CHARGE CODE
F.CHG/CODE	209	4	C		USER DEFINED CHARGE CODE
F.APPL/CODE	213	8	C		USER DEFINED APPLICATION CODE
F.PROG/CODE	221	4	C		USER DEFINED PROGRAM CODE
F.PGMR/CODE	221	4	C		USER DEFINED PROGRAM CODE
I..I RECORD	1	571	H		IDMS ACCOUNT RECORD
I.SYSTEM/ID	118	8	C		SYSTEM ID
I.RECORD/MTYPE	126	1	C		RECORD MAINTYPE
I.RECORD/STYPE	127	1	C		RECORD SUBTYPE
I.STATS/TYPE	128	2	B	NT	STATISTICS TYPE

Name	Pos	Len	Format	Suffixes	Description
I.SIGNON/ID	130	32	C		SIGN ON ID -CV
I.LOGICAL/TERMID	162	8	C		LOGICAL TERMINAL ID - CV
I.ACCT/INFO	170	16	C		USER INFORMATION FROM JOB CARD - ER
I.PWVS/SEQNO	186	2	B	NT	POWER SEQUENCE NUMBER - ER
I.PWVS/ACCT/INFO	188	14	C		POWER USER INFORMATION 1ST 14 BYTES - ER
I.TASK/CODE	202	8	C		TASK IDENTIFIER
I.IRESRVBIN	210	2	B		RESERVED BINARY FIELD
I.COND/INDS	212	3	C		CONDENSE OPTION INDICATORS
I.COND/RECS	215	4	P	SP	NUMBER OF RECORDS CONDENSED
I.PGMS/CALLED	219	8	P	SP	NUMBER OF PROGRAMS CALLED
I.PGMS/LOADED	227	8	P	SP	NUMBER OF PROGRAMS LOADED
I.TERM/READS	235	8	P	SP	NUMBER OF TERMINAL READS
I.TERM/WRITES	243	8	P	SP	NUMBER OF TERMINAL WRITES
I.TERM/ERRORS	251	4	B	SP	NUMBER OF TERMINAL I/O ERRORS
I.TASK/NMBR	255	4	B	SP	TASK NUMBER - NEGATIVE IF ERU DATA
I.PAGE/REQUESTS	259	8	P	SP	NUMBER OF PAGE REQUESTS
I.TEMPSTG/GETS	267	8	P	SP	NUMBER OF TEMP STORAGE GETS
I.TEMPSTG/PUTS	275	8	P	SP	NUMBER OF TEMP STORAGE PUTS
I.TEMPSTG/DELS	283	8	P	SP	NUMBER OF TEMP STORAGE DELS
I.QUEUE/READS	291	8	P	SP	NUMBER OF QUEUE READS
I.QUEUE/WRITES	299	8	P	SP	NUMBER OF QUEUE WRITES
I.QUEUE/DELETES	307	8	P	SP	NUMBER OF QUEUE DELETES
I.GETTIME/ REQUESTS	315	8	P	SP	NUMBER OF GETTIME REQUESTS
I.SETTIME/ REQUESTS	323	8	P	SP	NUMBER OF SETTIME REQUESTS
I.IO/CALLS	331	8	P	SP	NUMBER OF I/O CALLS
I.STACK/HWM	339	4	B	NT	STACK HIGH WATER MARK -MAX REQUESTS/TASK

Name	Pos	Len	Format	Suffixes	Description
I.USER/CPU	343	8	P	.4 MS SP	USER MODE CPU TIME
I.SYSTEM/CPU	351	8	P	.4 MS SP	SYSTEM MODE CPU TIME
I.TASK/WAIT	359	8	P	.4 MS SP	TASK WAIT TIME
I.PAGES/READ	367	8	P	SP	NUMBER OF PAGES READ
I.PAGES/WRITTEN	375	8	P	SP	NUMBER OF PAGES WRITTEN
I.PAGES/ REQUESTED	383	8	P	SP	NUMBER OF PAGES REQUESTED
I.CALC/NOFLOW	391	8	P	SP	CALCULATED RECORDS WITHOUT OVERFLOW
I.CALC/OVERFLOW	399	8	P	SP	CALCULATED RECORDS WITH OVERFLOW
I.VIA/NOFLOW	407	8	P	SP	VIA RECORDS WITHOUT OVERFLOW
I.VIA/OVERFLOW	415	8	P	SP	VIA RECORDS WITH OVERFLOW
I.RECORDS/ REQUESTD	423	8	P	SP	TOTAL RECORDS REQUESTED
I.RCRDS/ CURRENTRU	431	8	P	SP	TOTAL RECORDS CURRENT OF RU
I.FRAGMENTS/ STORED	439	8	P	SP	TOTAL FRAGMENTS STORED
I.RECORDS/ RELOCATD	447	8	P	SP	TOTAL RECORDS RELOCATED
I.LOCKS	455	8	P	SP	NUMBER OF LOCKS
I.SELECT/LOCKS	463	8	P	SP	NUMBER OF SELECT LOCKS
I.UPDATE/LOCKS	471	8	P	SP	NUMBER OF UPDATE LOCKS
I.STRG/HWM	479	4	B	NT	STORAGE HIGH WATER MARK
I.FREESTR/ REQUESTS	483	8	P	SP	FREE STORAGE REQUESTS
I.SYSSERV/ REQUESTS	491	8	P	SP	SYSTEM SERVICE REQUESTS
I.RESERVED/SMCPU	499	8	P	MS SP	SYSTEM-MODE CPU TIME IN 1/60 SECONDS
I.RESERVED/TWAIT	507	8	P	MS SP	TASK WAIT TIME IN 1/60 SECONDS
I.RESERVED/ UMCPU	515	8	P	MS SP	USER-MODE CPU TIME IN 1/60 SECONDS
I.USER/FIELD1	523	16	C		USER DATA FIELD
I.USER/FIELD2	539	5	P	SP	USER COUNT FIELD
I.USER/FIELD3	544	8	P	SP	USER TIME FIELD
I.DIV/CODE	552	4	C		DIVISION CODE

Name	Pos	Len	Format	Suffixes	Description
I.CHARGE/CODE	556	4	C		USER DEFINED CHARGE CODE
I.CHG/CODE	556	4	C		USER DEFINED CHARGE CODE
I.APPL/CODE	560	8	C		USER DEFINED APPLICATION CODE
I.PROG/CODE	568	4	C		USER DEFINED PROGRAM CODE
I.PGMR/CODE	568	4	C		USER DEFINED PROGRAM CODE
J..J RECORD	1	253	H		RJE SIGNFF RECORD
J.RJE/REMOTE/ID	118	3	C		REMOTE NAME-BSC
J.REMOTE NAME	118	8	C		REMOTE NAME-SNA
J.NETO/NAME	118	8	C		NODE NAME
J.LINE NAME	126	8	C		LOGICAL UNIT/LINE NAME
J.NETO/LINE	126	8	C		LINE NAME
J.PASSWORD	134	8	C		PASSWORD
J.NETO/LINEPASS	134	8	C		LINE PASSWORD
J.EXCP/COUNT	142	4	B	SP	NUMBER OF SIO
J.SIO/COUNT	142	4	B	SP	NUMBER OF SIO
J.NETO/EXCP	142	4	B	SP	TRANSMISSION COUNT (BSC ONLY)
J.LINE/ERROR	146	4	B	SP	NUMBER OF LINE ERRORS (BSC ONLY)
J.NETO/BUFFER	146	4	B	SP	BUFFERS RECEIVED (SNA ONLY)
J.TIMEOUT	150	4	B	SP	NUMBER OF READ TEXT TIME OUTS
J.INV/RESP	154	4	B	SP	NUMBER OF INVALID RESPONSES
J.LINE/ADDR	158	3	C		LINE ADAPTOR ADDRESS
J.NETO/LINEADR	158	3	C		LINE ADDRESS
J.SIGNOFF	161	2	C		SIGNOFF INDICATOR
J.NETO/CANCEL	161	2	C		CANCEL CODE
J.TERM/ERROR	163	2	B	SP	NUMBER OF TERMINAL ERRORS
J.NETO/TERR	163	2	B	SP	TERMINAL ERROR NUMBER
J.RJE/ACTIVE	165	8	P	.4 HS SP	RJE ACTIVE TIME IN SECONDS
J.RJE/IDLE	173	8	P	.4 HS SP	RJE IDLE TIME IN SECONDS
J.RJE/CONNECT	181	8	P	.4 HS SP	RJE CONNECT TIME IN SECONDS

Name	Pos	Len	Format	Suffixes	Description
J.RJE/USER/INFO	189	16	C		USER INFO FIELD FROM SIGNON COMMAND
J.USER/FIELD1	205	16	C		USER DATA FIELD
J.USER/FIELD2	221	5	P	SP	USER COUNT FIELD
J.USER/FIELD3	226	8	P	SP	USER TIME FIELD
J.DIV/CODE	234	4	C		DIVISION CODE
J.CHARGE/CODE	238	4	C		USER DEFINED CHARGE CODE
J.CHG/CODE	238	4	C		USER DEFINED CHARGE CODE
J.APPL/CODE	242	8	C		USER DEFINED APPLICATION CODE
J.PROG/CODE	250	4	C		USER DEFINED PROGRAM CODE
J.PGMR/CODE	250	4	C		USER DEFINED PROGRAM CODE
K..K RECORD	1	275	H		
K.POWER/ID	118	1	C		POWER ID
K.NETK/JES2/JES3	118	1	C		JES2/JES3/ POWER ID
K.JOB/SFX	119	1	C		JOB SUFFIX NUMBER
K.NETK/JOB/SFX	119	1	C		JOB SUFFIX NUMBER
K.ORIG/NAME	120	8	C		NODE NAME (ORIGIN)
K.NETK/NAME	120	8	C		NODE NAME (ORIGIN)
K.EXEC/NAME	128	8	C		EXECUTION NODE NAME
K.NETK/EXEC	128	8	C		EXECUTION NODE NAME
K.DEST/NAME	136	8	C		DESTINATION NODE NAME
K.NETK/DEST	136	8	C		DESTINATION NODE NAME
K.JOBN/ORIG	144	2	B	NT	ORIGINAL JOB NUMBER
K.NETK/JOBN/ ORIG	144	2	B	NT	ORIGINAL JOB NUMBER
K.JOBN/CURR	146	2	B	NT	CURRENT JOB NUMBER
K.NETK/JOBN/ CURR	146	2	B	NT	CURRENT JOB NUMBER
K.ORIG/USERID	148	8	C		ORIGINAL USER IDENTIFIER
K.NETK/USER ORIG	148	8	C		ORIGINAL USER IDENTIFIER
K.NETK/DEV NAME	148	8	C		
K.USER DEST	156	8	C		DESTINATION USER ID

Name	Pos	Len	Format	Suffixes	Description
K.NETK/USER DEST	156	8	C		DESTINATION USER ID
K.TP COUNT	164	4	B	SP	RECORD COUNT
K.NETK/TP COUNT	164	4	B	SP	RECORD COUNT
K.CB COUNT	168	4	B	SP	COMPRESSED REC/CONTROL REC COUNT
K.NETK/CB COUNT	168	4	B	SP	COMPRESSED REC/CONTROL REC COUNT
K.USER INFO	172	16	C		USER ACCOUNTING INFORMATION
K.NETK/USER INFO	172	16	C		USER ACCOUNTING INFORMATION
K.ACCOUNT/NMBR	188	8	C		NETWORK ACCOUNT NUMBER
K.NETK/ACCOUNT	188	8	C		NETWORK ACCOUNT NUMBER
K.DEPT	196	8	C		DEPARTMENT NUMBER
K.NETK DEPT	196	8	C		DEPARTMENT NUMBER
K.BLDG	204	8	C		BUILDING NUMBER
K.NETK BLDG	204	8	C		BUILDING NUMBER
K.LOC	212	8	C		LOCATION NUMBER
K.NETK LOC	212	8	C		LOCATION NUMBER
K.ADJ/PATH	220	8	C		ADJACENT NODE NAME
K.NETK/ADJ PATH	220	8	C		ADJACENT NODE NAME
K.NETK/CANCEL	228	2	C		CANCEL CODE
K.LINADR	230	3	C		LINE ADDRESS OF SNA
K.NETK/LINADR	230	3	C		LINE ADDRESS OF SNA
K.PRTY	233	1	C		PRIORITY
K.NETK/PRTY	233	1	C		PRIORITY
K.CLASS	234	1	C		CLASS
K.NETK/CLASS	234	1	C		CLASS
K.QUE/ID	235	1	C		QUEUE RECORD IDENTIFIER
K.NETK/QUE/ID	235	1	C		QUEUE RECORD IDENTIFIER
K.COPIES	236	2	B		NUMBER OF COPIES
K.NETK/COPIES	236	2	B		NUMBER OF COPIES
K.JOB/TYPE	238	2	C		JOB TYPE INDICATOR

Name	Pos	Len	Format	Suffixes	Description
K.NETK/JOB/TYPE	238	2	C		JOB TYPE INDICATOR
K.DIV/CODE	240	4	C		DIVISION CODE
K.CHARGE/CODE	244	4	C		USER DEFINED CHARGE CODE
K.CHG/CODE	244	4	C		USER DEFINED CHARGE CODE
K.APPL/CODE	248	8	C		USER DEFINED APPLICATION CODE
K.PROG/CODE	256	4	C		USER DEFINED PROGRAM CODE
K.PGMR/CODE	256	4	C		USER DEFINED PROGRAM CODE
K.NETWORK/ACCT NO	260	8	C		NETWORK ACCOUNT NUMBER
K.RECS/INSERTED	268	4	B	L6 SP	OUTEXIT RECORDS INSERTED
K.RECS/DELETED	272	4	B	L6 SP	OUTEXIT RECORDS DELETED
L..L RECORD	1	376	H		OUTPUT WRITER RECORD
L.PROGRAM/NAME	118	8	C		EITHER PRINTER OR PUNCH
L.LINES/PRINTED	126	4	B	L7 SP	NUMBER OF LINES PRINTED
L.CARDS/PUNCHED	130	4	B	L7 SP	NUMBER OF CARDS PUNCHED
L.PRINT/TIME	134	8	P	.4 HM SP	TIME TAKEN FOR PRINTING
L.PUNCH/TIME	142	8	P	.4 HM SP	TIME TAKEN FOR PUNCHING
L.FORM/NO	150	4	C		FORM NUMBER
L.FROM/USER	154	8	C		FROM USER
L.SYSOUT/CL	162	1	C		SYSOUT CLASS
L.SYSOUT/CLASS	162	1	C		SYSOUT CLASS
L.PAGE/COUNT	163	4	B	L6 SP	NUMBER OF PAGES PRINTED
L.OUTPUT/PRTY	167	2	B	NT	OUTPUT PRIORITY
L.JOB/PRTY	169	2	B	NT	JOB PRIORITY
L.JOB/FROM/USER	171	8	C		JOB FROM USER
L.RJE/IN	171	3	C		INPUT REMOTE DEVICE DECIMAL
L.JOB/TO/USER	179	8	C		JOB TO USER
L.RJE/OUT	179	3	C		OUTPUT REMOTE DEVICE IN DECIMAL
L.RUN/TYPE	187	1	C		RUN TYPE - P,T,C,N,R OR U
L.SUB/RUN/TYPE	188	1	C		SUB RUN TYPE

Name	Pos	Len	Format	Suffixes	Description
L.DOS/JOBNAME	189	8	C		DOS JOB NAME
L.DOS DESC	197	16	C		DOS DESCRIPTION
L.DOS/ DESCRIPTION	197	16	C		DOS DESCRIPTION
L.PWVS DESC	213	16	C		POWER DESCRIPTION
L.PWVS/ DESCRIPTION	213	16	C		POWER DESCRIPTION
L.PWVS/JOBNAME	229	8	C		POWER JOB NAME
L.AA/PWVS/ JOBNAME	229	8	C		FORCE DOS FORMATTING OF OUTPUT RECORD
L.PWVS/SEQNO	237	2	B	NT	POWER SEQUENCE NUMBER
L.PWVS/ABEND	239	1	C		POWER JOB ABEND INDICATOR
L.PWVS/CANCEL	240	4	C		POWER CANCEL CODE
L.PWVS/TRKS	244	4	B	SP	NUMBER OF TRACKS USED FOR STORAGE
L.DEV/ADDR	248	3	C		OUTPUT DEVICE ADDRESS, SNA OR GSP
L.JOB COMP CODE	251	4	C		COMPLETION CODE
L.JOB TERM CODE	255	1	C		TERMINATION CODE
L.PWVS/SUFFIX	256	2	B	NT	JOB SUFFIX NUMBER
L.COPIES	258	2	B		NUMBER OF COPIES
L.EXTRA/LINES	260	4	B	SP	NUMBER OF EXTRA LINES
L.EXTRA/PAGES	264	4	B	SP	NUMBER OF EXTRA PAGES
L.EXTRA/CARDS	268	4	B	SP	NUMBER OF EXTRA CARDS
L.PART	272	2	C		PARTITION ID
L.FLSH/ID	274	4	C		3800 FLASH IDENTIFIER
L.COPY/GROUP	278	8	H		COPY GROUPINGS - 3800 ONLY
L.COPY/GROUP1	278	1	B	NT	COPY GROUP FIELD 1
L.COPY/GROUP2	279	1	B	NT	COPY GROUP FIELD 2
L.COPY/GROUP3	280	1	B	NT	COPY GROUP FIELD 3
L.COPY/GROUP4	281	1	B	NT	COPY GROUP FIELD 4
L.COPY/GROUP5	282	1	B	NT	COPY GROUP FIELD 5
L.COPY/GROUP6	283	1	B	NT	COPY GROUP FIELD 6

Name	Pos	Len	Format	Suffixes	Description
L.COPY/GROUP7	284	1	B	NT	COPY GROUP FIELD 7
L.COPY/GROUP8	285	1	B	NT	COPY GROUP FIELD 8
L.OWN/NODE	286	8	C		NAME OF OWN NODE IN THE NETWORK
L.TO/USER	294	8	C		ORIGINATING TO NODE
L.ORIG/NODE	302	8	C		ORIGINATING FROM NODE
L.ORIG/SEQNO	310	2	B	NT	ORIGINAL JOB NUMBER
L.USER/FIELD1	312	16	C		USER DATA FIELD
L.USER/FIELD2	328	5	P	SP	USER COUNT FIELD
L.USER/FIELD3	333	8	P	SP	USER TIME FIELD
L.DIV/CODE	341	4	C		DIVISION CODE
L.CHARGE/CODE	345	4	C		USER DEFINED CHARGE CODE
L.CHG/CODE	345	4	C		USER DEFINED CHARGE CODE
L.APPL/CODE	349	8	C		USER DEFINED APPLICATION CODE
L.PROG/CODE	357	4	C		USER DEFINED PROGRAM CODE
L.PGMR/CODE	357	4	C		USER DEFINED PROGRAM CODE
L.NETWORK/ACCT NO	361	8	C		NETWORK ACCOUNT NUMBER
L.RECS/INSERTED	369	4	B	L6 SP	OUTEXIT RECORDS INSERTED
L.RECS/DELETED	373	4	B	L6 SP	OUTEXIT RECORDS DELETED
M..M RECORD	1	285	H		MISCL RECORD - CREATED BY THE USER
M.ACCT INFO	118	25	C		ACCOUNTING INFORMATION
M.DIV/CODE	143	4	C		DIVISION CODE
M.CHARGE/CODE	147	4	C		USER DEFINED CHARGE CODE
M.CHG/CODE	147	4	C		USER DEFINED CHARGE CODE
M.APPL/CODE	151	8	C		USER DEFINED APPLICATION CODE
M.PGMR/CODE	159	4	C		USER DEFINED PROGRAM CODE
M.PROG/CODE	159	4	C		USER DEFINED PROGRAM CODE
M.MISC/CODE	163	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/COUNT	165	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE

Name	Pos	Len	Format	Suffixes	Description
M.MISC/TIME	169	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/CHARGE	177	4	B	.2 L6 SP	USER DEFINED CHARGE FOR CHARGE TYPE CODE
M.MISC/CODE1	181	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME1	183	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT1	191	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE2	195	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME2	197	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT2	205	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE3	209	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME3	211	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT3	219	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE4	223	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME4	225	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT4	233	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE5	237	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME5	239	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT5	247	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.TOTAL/CHRG	251	6	P	.2 SP	TOTAL CHARGE
M.USER/FIELD1	257	16	C		USER DATA FIELD
M.USER/FIELD2	273	5	P	SP	USER COUNT FIELD
M.USER/FIELD3	278	8	P	SP	USER TIME FIELD
N..N RECORD	1	201	C		NETWORK RECORD

Name	Pos	Len	Format	Suffixes	Description
N.NODE/ID	118	8	C		NODE IDENTIFIER
N.NODE/PWD	126	8	C		NODE PASSWORD
N.LINE/PWD	134	8	C		LINE PASSWORD
N.INVALID/RESP	142	8	P	SP	INVALID RESPONSES PER SESSION
N.PWVS/ABEND	150	1	C		POWER JOB ABEND INDICATOR
N.PWVS/CANCEL	151	4	C		POWER CANCEL CODE
N.LINE/ADDR	155	3	C		LINE ADDRESS (CUU) OR SNA
N.TRANS/COUNT	158	8	P	SP	TRANSMISSION COUNT PER SESSION
N.TMOUT/COUNT	166	4	B	SP	TIMEOUT COUNT PER SESSION (BSC)
N.ERROR/COUNT	170	4	B	SP	ERROR COUNT PER SESSION (BSC)
N.BUFFS/RCVD	174	8	P	SP	BUFFERS RECEIVED DURING SESSION (SNA)
N.DIV/CODE	182	4	C		DIVISION CODE
N.CHARGE/CODE	186	4	C		USER DEFINED CHARGE CODE
N.CHG/CODE	186	4	C		USER DEFINED CHARGE CODE
N.APPL/CODE	190	8	C		USER DEFINED APPLICATION CODE
N.PROG/CODE	198	4	C		USER DEFINED PROGRAM CODE
N.PGMR/CODE	198	4	C		USER DEFINED PROGRAM CODE
P..O RECORD	1	406	H		INTERFACE RECORD
P.ID/FIELD1	118	2	C		ID FIELD 1
P.USER/CODE	118	2	C		UNIVERSE USERS 2-DIGIT CODE
P.ID/FIELD2	120	1	C		ID FIELD 2
P.USER/TYPE	120	1	C		UNIVERSE USER TYPE 1=DBA, 0=NONDBA
P.ID/FIELD3	121	8	C		ID FIELD 3
P.ID/FIELD4	129	16	C		ID FIELD 4
P.ID/FIELD41	129	4	C		ID FIELD 4 L4
P.ID/FIELD42	133	4	C		ID FIELD 4 L4
P.ID/FIELD43	137	4	C		ID FIELD 4 L4

Name	Pos	Len	Format	Suffixes	Description
P.ID/FIELD44	141	4	C		ID FIELD 4 L4
P.ID/FIELD5	145	16	C		ID FIELD 5
P.ID/FIELD51	145	4	C		ID FIELD 5 L4
P.ID/FIELD52	149	4	C		ID FIELD 5 L4
P.ID/FIELD53	153	4	C		ID FIELD 5 L4
P.ID/FIELD54	157	4	C		ID FIELD 5 L4
P.ID/FIELD6	161	4	C		ID FIELD 6
P.ID/FIELD7	165	4	C		ID FIELD 7
P.ID/FIELD8	169	4	C		ID FIELD 8
P.ID/FIELD9	173	2	H		ID FIELD 9
P.ID/FIELD10	175	2	B		ID FIELD 10
P.TIME/FIELD1	177	8	P	.4 MS SP	TIME FIELD 1
P.CPU/CLEAR	177	8	P	.4 MS SP	UNIVERSE CPU TIME SINCE LAST CLEAR
P.TIME/FIELD2	185	8	P	.4 MS SP	TIME FIELD 2
P.CPU/THRESH	185	8	P	.4 MS SP	UNIVERSE CPU TIME SINCE LAST THRESH.
P.TIME/FIELD3	193	8	P	.4 MS SP	TIME FIELD 3
P.TIME/FIELD4	201	8	P	.4 MS SP	TIME FIELD 4
P.TIME/FIELD5	209	8	P	.4 MS SP	TIME FIELD 5
P.TIME/FIELD6	217	8	P	.4 MS SP	TIME FIELD 6
P.TIME/FIELD7	225	8	P	.4 MS SP	TIME FIELD 7
P.TIME/FIELD8	233	8	P	.4 MS SP	TIME FIELD 8
P.TIME/FIELD9	241	8	P	.4 MS SP	TIME FIELD 9
P.TIME/FIELD10	249	8	P	.4 MS SP	TIME FIELD 10
P.COUNT/FIELD1	257	4	B	SP	COUNT FIELD 1
P.IOS/REQUESTED	257	4	B	SP	UNIVERSE IO REQS SINCE LAST CLEAR
P.COUNT/FIELD2	261	4	B	SP	COUNT FIELD 2
P.IOS/CLEAR	261	4	B	SP	UNIVERSE IOS SINCE LAST CLEAR
P.COUNT/FIELD3	265	4	B	SP	COUNT FIELD 3

Name	Pos	Len	Format	Suffixes	Description
P.IOS/THRESH	265	4	B	SP	UNIVERSE IOS SINCE LAST THRESHOLD
P.COUNT/FIELD4	269	4	B	SP	COUNT FIELD 4
P.CPU/WARNINGS	269	4	B	SP	UNIVERSE COUNT OF CPU WARNINGS
P.COUNT/FIELD5	273	4	B	SP	COUNT FIELD 5
P.IO/WARNINGS	273	4	B	SP	UNIVERSE COUNT OF IO WARNINGS
P.COUNT/FIELD6	277	4	B	SP	COUNT FIELD 6
P.RES/WAITS	277	4	B	SP	UNIVERSE NUMBER OF RESOURCE WAITS
P.COUNT/FIELD7	281	4	B	SP	COUNT FIELD 7
P.DD/COMMANDS	281	4	B	SP	UNIVERSE DATA DEF COMMANDS
P.COUNT/FIELD8	285	4	B	SP	COUNT FIELD 8
P.DD/UPDATES	285	4	B	SP	UNIVERSE DATA DEF UPDATES SINCE CLR
P.COUNT/FIELD9	289	4	B	SP	COUNT FIELD 9
P.INT/VIOLS	289	4	B	SP	UNIVERSE NUMBER OF INTEG. VIOLATIONS
P.COUNT/FIELD10	293	4	B	SP	COUNT FIELD 10
P.LUW/ROLLBKS	293	4	B	SP	UNIVERSE NUMBER OF LUW ROLLBACKS
P.COUNT/FIELD11	297	4	B	SP	COUNT FIELD 11
P.USER/DEADLKS	297	4	B	SP	UNIVERSE NUMBER OF USER DEADLOCKS
P.COUNT/FIELD12	301	4	B	SP	COUNT FIELD 12
P.USER/FORCES	301	4	B	SP	UNIVERSE NUMBER OF USER FORCES
P.COUNT/FIELD13	305	4	B	SP	COUNT FIELD 13
P.COUNT/FIELD14	309	4	B	SP	COUNT FIELD 14
P.COUNT/FIELD15	313	4	B	SP	COUNT FIELD 15
P.COUNT/FIELD16	317	4	B	SP	COUNT FIELD 16
P.COUNT/FIELD17	321	4	B	SP	COUNT FIELD 17

Name	Pos	Len	Format	Suffixes	Description
P.COUNT/FIELD18	325	4	B	SP	COUNT FIELD 18
P.COUNT/FIELD19	329	4	B	SP	COUNT FIELD 19
P.COUNT/FIELD20	333	4	B	SP	COUNT FIELD 20
P.MISC/FIELD1	337	1	H		MISC FIELD 1
P.EDITOR/PERM	337	1	H		UNIVERSE PERMISSION TO USE EDITOR
P.MISC/FIELD2	338	1	H		MISC FIELD 2
P.UGEN/PERM	338	1	H		UNIVERSE PERMISSION TO USE UGEN
P.MISC/FIELD3	339	1	H		MISC FIELD 3
P.UPREP/PERM	339	1	H		UNIVERSE PERMISSION TO USE UPREP
P.MISC/FIELD4	340	1	H		MISC FIELD 4
P.LUW/PERM	340	1	H		UNIVERSE PERMISSION TO USE LUW
P.MISC/FIELD5	341	1	H		MISC FIELD 5
P.FUNC/PERM	341	1	H		UNIVERSE PERMISSION TO USE FUNCTIONS
P.MISC/FIELD6	342	1	H		MISC FIELD 6
P.OVER/PERM	342	1	H		UNIVERSE PERM. TO OVERRIDE IO/CPU THRS
P.MISC/FIELD7	343	1	H		MISC FIELD 7
P.MISC/FIELD8	344	1	H		MISC FIELD 8
P.MISC/FIELD9	345	1	H		MISC FIELD 9
P.MISC/FIELD10	346	1	H		MISC FIELD 10
P.MISC/FIELD11	347	4	B		MISC FIELD 11
P.CPU/LIMIT	347	4	B		UNIVERSE CPU SECS BEFORE WARNING
P.MISC/FIELD12	351	4	B		MISC FIELD 12
P.IO/LIMIT	351	4	B		UNIVERSE IOS BEFORE WARNING
P.MISC/FIELD13	355	4	B		MISC FIELD 13
P.MISC/FIELD14	359	4	B		MISC FIELD 14
P.MISC/FIELD15	363	4	B		MISC FIELD 15

Name	Pos	Len	Format	Suffixes	Description
P.MISC/FIELD16	367	4	B		MISC FIELD 16
P.MISC/FIELD17	371	4	B		MISC FIELD 17
P.MISC/FIELD18	375	4	B		MISC FIELD 18
P.MISC/FIELD19	379	4	B		MISC FIELD 19
P.MISC/FIELD20	383	4	B		MISC FIELD 20
P.DIV/CODE	387	4	C		DIVISION CODE
P.CHARGE/CODE	391	4	C		USER DEFINED CHARGE CODE
P.CHG/CODE	391	4	C		USER DEFINED CHARGE CODE
P.APPL/CODE	395	8	C		USER DEFINED APPLICATION CODE
P.PROG/CODE	403	4	C		USER DEFINED PROGRAM CODE
P.PGMR/CODE	403	4	C		USER DEFINED PROGRAM CODE
R..R RECORD	1	311	H		READER RECORD
R.JOB/CL	118	1	C		POWER INPUT CLASS
R.JOB/CLASS	118	1	C		POWER INPUT CLASS
R.JOB/PRTY	119	2	B	L4 NT	POWER INPUT PRIORITY
R.JOB/FROM/USER	121	8	C		JOB FROM USER
R.RJE/IN	121	3	C		INPUT REMOTE DEVICE IN DECIMAL
R.JOB/TO/USER	129	8	C		JOB TO USER
R.RJE/OUT	129	3	C		OUTPUT REMOTE DEVICE IN DECIMAL
R.RUN/TYPE	137	1	C		RUN TYPE - P,T,C,N,R OR U
R.SUB/RUN/TYPE	138	1	C		SUB RUN TYPE
R.DOS/JOBNAME	139	8	C		DOS JOB NAME
R.DOS DESC	147	16	C		DOS DESCRIPTION
R.DOS/ DESCRIPTION	147	16	C		DOS DESCRIPTION
R.PWVS DESC	163	16	C		POWER DESCRIPTION
R.PWVS/ DESCRIPTION	163	16	C		POWER DESCRIPTION
R.PWVS/ JOBNAME	179	8	C		POWER JOB NAME
R.AA/PWVS/ JOBNAME	179	8	C		FORCE DOS FORMATTING OF OUTPUT RECORD

Name	Pos	Len	Format	Suffixes	Description
R.PWVS/SEQNO	187	2	B	NT	POWER SEQUENCE NUMBER
R.PWVS/ABEND	189	1	C		POWER JOB ABEND INDICATOR
R.PWVS/CANCEL	190	4	C		POWER CANCEL CODE
R.PWVS/TRKS	194	4	B	SP	NUMBER OF TRACKS USED FOR STORAGE
R.DEV/ADDR	198	3	C		READER DEVICE ADDRESS, SNA OR PSP
R.PART	201	2	C		PARTITION ID
R.CARD/IMAGE/ RCDS	203	4	B	SP	NUMBER OF CARD IMAGE RECORDS
R.RDR/DURATION	207	8	P	.4 HS	READER DURATION TIME
R.RESERVED PACK1	215	8	P		RESERVED PACKED FIELD 1
R.RESERVED PACK2	223	8	P		RESERVED PACKED FIELD 2
R.RESERVED PACK3	231	8	P		RESERVED PACKED FIELD 3
R.OWN/NODE	239	8	C		NAME OF OWN NODE IN THE NETWORK
R.FROM/USER	247	8	C		ORIGINATING- USER ID
R.USER/FIELD1	255	16	C		USER DATA FIELD
R.USER/FIELD2	271	5	P	SP	USER COUNT FIELD
R.USER/FIELD3	276	8	P	SP	USER TIME FIELD
R.DIV/CODE	284	4	C		DIVISION CODE
R.CHARGE/CODE	288	4	C		USER DEFINED CHARGE CODE
R.CHG/CODE	288	4	C		USER DEFINED CHARGE CODE
R.APPL/CODE	292	8	C		USER DEFINED APPLICATION CODE
R.PROG/CODE	300	4	C		USER DEFINED PROGRAM CODE
R.PGMR/CODE	300	4	C		USER DEFINED PROGRAM CODE
R.NETWORK/ACCT NO	304	8	C		NETWORK ACCOUNT NUMBER
S..S RECORD	1	163	C		SPOOL ACCESS CONNECT
S.XPCC/APPLID	118	8	C	SP	XPCC APPLICATION IDENTIFIER
S.NO MSGS/RETURNED	126	4	B	SP	NO. MSGS RETURNED IN RESPONSE TO CTL/PUT
S.NO CTL/REQUESTS	130	4	B	SP	NO. OF CTL REQUESTS

Name	Pos	Len	Format	Suffixes	Description
S.CON/TERM/CODE	134	1	H		CONNECTION- TERMINATION CODE
S.DEVICE/NAME	135	8	C		DEVICE NAME - AS DEFINED TO SUBSYSTEM
S.ACCOUNT/RECID	143	1	C		ACCOUNT- RECORD IDENTIFIER
S.DIV/CODE	144	4	C		DIVISION CODE
S.CHARGE/CODE	148	4	C		USER DEFINED CHARGE CODE
S.CHG/CODE	148	4	C		USER DEFINED CHARGE CODE
S.APPL/CODE	152	8	C		USER DEFINED APPLICATION CODE
S.PROG/CODE	160	4	C		USER DEFINED PROGRAM CODE
S.PGMR/CODE	160	4	C		USER DEFINED PROGRAM CODE
U..U RECORD	1	130	H		UNIT/DEVICE ACTIVITY RECORD
U.DEV/ADDR	118	3	C		DEVICE ADDRESS - CUU
U.DEVADDR/ RESERVED	118	3	C		DEVICE ADDRESS - CUU
U.DEV/CL	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
U.DEV/CLASS	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
U.UNIT/TYPE	122	1	P	NT	UNIT TYPE -
U.UNITTYPE	122	1	P	NT	UNIT TYPE -
U.DEV/TYPE	123	4	C		DVC TYPE-EG 3420-NOT CURRENTLY AVAILABLE
U.DEVICE/SIOS	127	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
U.REQUESTS/SERV	127	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
U.NUMSIOS/ RESERVED	127	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
V..V RECORD	1	264	H		VM CONSOLIDATED RECORD
V.BATCH/ INDICATOR	118	1	C		B=BATCH MACHINE
V.BATCH/JOB/ NAME	119	8	C		BATCH JOB NAME (IF BATCH)
V.ACCOUNT/ NUMBER	127	16	C		VM ACCOUNT NUMBER
V.ACCT/NO	127	16	C		VM ACCOUNT NUMBER (08)
V.USER/INFO	127	16	C		USER INFO FROM USER ACCNT TABLE (16)

Name	Pos	Len	Format	Suffixes	Description
V.PROJECT/CODE	143	8	C		PROJECT ACCOUNTING PROJECT CODE
V.PRODUCT/NAME	151	8	C		PRODUCT ACCOUNTING PRODUCT NAME
V.PRODUCT/ PRIORITY	159	1	C		PRODUCT ACCOUNTING PRODUCT PRIORITY
V.CMS/ABEND/ CODE	160	3	C		RIGHTMOST 3 BYTES OF CMS ABEND CODE
V.TOTAL/CPU/TIME	163	8	P	.4 MS SP	
V.OVHD/CPU/TIME	171	8	P	.4 MS SP	
V.VIRT/CPU/TIME	179	8	P	.4 MS SP	
V.PAGE/READS	187	4	B	L7 SP	NUMBER OF PAGE-READ OPERATIONS
V.PAGE/WRITES	191	4	B	L7 SP	NUMBER OF PAGE-WRITE OPERATIONS
V.TOTAL/SIO	195	4	B	L7 SP	TOTAL SIOS
V.NON/SPOOL/SIO	199	4	B	L7 SP	NUMBER OF NON-SPOOLED DEVICE SIOS
V.CARDS/PUNCHED	203	4	B	L7 SP	NUMBER OF VIRTUAL CARDS PUNCHED
V.LINES/PRINTED	207	4	B	L7 SP	NUMBER OF VIRTUAL LINES PRINTED
V.CARDS/READ	211	4	B	L7 SP	NUMBER OF VIRTUAL CARDS READ
V.NO/TAPES/USED	215	2	P		NUMBER OF TAPE DRIVES USED
V.NO/DISKS/USED	217	2	P		NUMBER OF DISK DRIVES USED
V.NO/OTHER/USED	219	2	P		NUMBER OF OTHER DEVICES USED
V.TAPE/CONN/TIME	221	8	P	.4 MS SP	
V.NON/TAPE/TIME	229	8	P	.4 MS SP	
V.TDISK/CYL/SECS	237	8	P	SP	T-DISK CYLINDERS * SECONDS OF CONN TIME
V.DIV/CODE	245	4	C		DIVISION CODE
V.CHARGE/CODE	249	4	C		USER DEFINED CHARGE CODE
V.CHG/CODE	249	4	C		USER DEFINED CHARGE CODE
V.APPL/CODE	253	8	C		USER DEFINED APPLICATION CODE

Name	Pos	Len	Format	Suffixes	Description
V.PROG/CODE	261	4	C		USER DEFINED PROGRAM CODE
V.PGMR/CODE	261	4	C		USER DEFINED PROGRAM CODE
X..X RECORD	1	271	C		SPOOL ACCESS OPERATION
X.PWVS DESC	118	16	C		POWER DESCRIPTION FIELD
X.PWVS/ DESCRIPTION	118	16	C		POWER DESCRIPTION FIELD
X.PWVS/SEQNO	134	2	B	NT	VSE/POWER JOB SEQUENCE NUMBER
X.PWVS/CANCEL	136	4	C		POWER CANCEL CODE
X.PWVS/ABEND	140	1	C		POWER ABEND INDICATOR
X.REQ/TYPE	141	1	C		REQUEST TYPE (G=GET REQUEST P=PUT)
X.QUEUE/TYPE	142	1	C		QUEUE TYPE (R=RDR, L=LST, P=PUN)
X.JOB/SUFFIX	143	1	B	NT	JOB SUFFIX (SEGMENT) NUMBER
X.CLASS	144	1	C		CLASS OF JOB OUTPUT
X.PRTY	145	1	C		PRIORITY
X.DISP	146	1	C		DISPOSITION
X.COPIES	147	1	B	NT	NUMBER OF COPIES
X.COPY/ GROUPINGS	148	8	H		COPY GROUPINGS
X.COPY/GROUP1	148	1	B	NT	FIRST COPY GROUP FIELD
X.COPY/GROUP2	149	1	B	NT	SECOND COPY GROUP FIELD
X.COPY/GROUP3	150	1	B	NT	THIRD COPY GROUP FIELD
X.COPY/GROUP4	151	1	B	NT	FOURTH COPY GROUP FIELD
X.COPY/GROUP5	152	1	B	NT	FIFTH COPY GROUP FIELD
X.COPY/GROUP6	153	1	B	NT	SIXTH COPY GROUP FIELD
X.COPY/GROUP7	154	1	B	NT	SEVENTH COPY GROUP FIELD
X.COPY/GROUP8	155	1	B	NT	EIGHT COPY GROUP FIELD
X.PWVS/TRKS	156	2	B		NUMBER OF TRACKS FOR STORAGE
X.ORIG/SEQNO	158	2	B	NT	ORIGINAL SEQUENCE NUMBER
X.RECORDS	160	4	B		NUMBER OF RECORDS
X.EXTRA/RECS	164	4	B		NUMBER OF EXTRA RECORDS

Name	Pos	Len	Format	Suffixes	Description
X.LINES/CARDS	168	4	B		LINES/CARDS
X.XTRA/LINES/ CARDS	172	4	B		EXTRA LINES/CARDS
X.NO/PAGES	176	4	B		NUMBER OF PAGES
X.EXTRA/PAGES	180	4	B		NUMBER OF EXTRA PAGES
X.FORM/IDENT	184	8	C		FORMS ID
X.FLSH/ID	192	4	C		3800 FLASH IDENTIFIER
X.DEST/NODE	196	8	C		DESTINATION NODE
X.DEST/USER	204	8	C		DESTINATION USER
X.REQ/USER	212	8	C		REQUESTING USER ID
X.REQ/APPLID	220	8	C		REQUESTING APPLICATION ID
X.LOCAL/NODE	228	8	C		REQUESTING APPLICATION ID
X.DIV/CODE	236	4	C		DIVISION CODE
X.CHARGE/CODE	240	4	C		USER DEFINED CHARGE CODE
X.CHG/CODE	240	4	C		USER DEFINED CHARGE CODE
X.APPL/CODE	244	8	C		USER DEFINED APPLICATION CODE
X.PROG/CODE	252	4	C		USER DEFINED PROGRAM CODE
X.PGMR/CODE	252	4	C		USER DEFINED PROGRAM CODE
X.NETWORK/ACCT NO	256	8	C		NETWORK ACCOUNT NUMBER
X.RECS/INSERTED	264	4	B	L6 SP	OUTEXIT RECORDS INSERTED
X.RECS/DELETED	268	4	B	L6 SP	OUTEXIT RECORDS DELETED
Z..Z RECORD	1	156	H		IPL RECORD
Z.PWR/VER	118	4	C		POWER VERSION/ MODIFICATION LEVEL
Z.PWR/LEV	122	4	C		POWER LEVEL INDICATOR
Z.PART/SIZE	126	4	B	NT	POWER PARTITION SIZE
Z.GETVS/SIZE	130	4	B	NT	POWER PARTITION GETVIS SIZE
Z.REAL/SIZE	134	4	B	NT	RESERVED PROCESSOR REAL STORAGE SIZE
Z.PWR/PART	138	2	C		POWER PARTITION ID
Z.PWR/FLAGS	140	4	C		POWER FEATURE FLAGS
Z.DFILE/EXTNTS	144	1	B	NT	NUMBER OF DATA FILE EXTENTS

Name	Pos	Len	Format	Suffixes	Description
Z.DFILE/TRACKS	145	4	B	NT	NUMBER OF DATA FILE TRACKS/BLOCKS
Z.QFILE/TRACKS	149	4	B	NT	NUMBER OF QUEUE FILE TRACKS/BLOCKS
Z.AFILE/TRACKS	153	4	B	NT	NUMBER OF ACCOUNT FILE TRACKS/BLOCKS
1..1 RECORD	1	185	H		VM USERID USAGE RECORD
1.ACCOUNT/ NUMBER	118	8	C		VM ACCOUNT NUMBER
1.ACCT/NO	118	8	C		VM ACCOUNT NUMBER
1.OVHD/CPU/TIME	126	8	P	.4 MS SP	
1.VIRT/CPU/TIME	134	8	P	.4 MS SP	
1.PAGE/READS	142	4	B	SP	NUMBER OF PAGE-READ OPERATIONS
1.PAGE/WRITES	146	4	B	SP	NUMBER OF PAGE-WRITE OPERATIONS
1.NON/SPOOL/SIO	150	4	B	SP	NUMBER OF NON-SPOOLED DEVICE SIOS
1.CARDS/PUNCHED	154	4	B	SP	NUMBER OF VIRTUAL CARDS PUNCHED
1.LINES/PRINTED	158	4	B	SP	NUMBER OF VIRTUAL LINES PRINTED
1.CARDS/READ	162	4	B	SP	NUMBER OF VIRTUAL CARDS READ
1.DIV/CODE	166	4	C		DIVISION CODE
1.CHARGE/CODE	170	4	C		USER DEFINED CHARGE CODE
1.CHG/CODE	170	4	C		USER DEFINED CHARGE CODE
1.APPL/CODE	174	8	C		USER DEFINED APPLICATION CODE
1.PROG/CODE	182	4	C		USER DEFINED PROGRAM CODE
1.PGMR/CODE	182	4	C		USER DEFINED PROGRAM CODE
2..2 RECORD	1	156	H		VM DEDICATED DEV AND T-DISK RECORD
2.ACCOUNT/ NUMBER	118	8	C		VM ACCOUNT NUMBER
2.ACCT/NO	118	8	C		VM ACCOUNT NUMBER
2.DEV/CLASS	126	1	C		VM DEVICE CLASS CODE

Name	Pos	Len	Format	Suffixes	Description
2.DEV/CL	126	1	C		VM DEVICE CLASS CODE
2.DEVICE/TYPE	127	4	C		VM DEVICE TYPE
2.MODEL	131	1	H		VM DEVICE MODEL CODE
2.FEATURE	132	1	H		VM DEVICE FEATURE CODE
2.NO OF/CYLS	133	4	B	SP	NUMBER OF CYLS/SETS OF 100 BLOCKS
2.NO OF/BLOCKS	133	4	B	SP	NUMBER OF CYLS/SETS OF 100 BLOCKS
2.DIV/CODE	137	4	C		DIVISION CODE
2.CHARGE/CODE	141	4	C		USER DEFINED CHARGE CODE
2.CHG/CODE	141	4	C		USER DEFINED CHARGE CODE
2.APPL/CODE	145	8	C		USER DEFINED APPLICATION CODE
2.PROG/CODE	153	4	C		USER DEFINED PROGRAM CODE
2.PGMR/CODE	153	4	C		USER DEFINED PROGRAM CODE
3..3 RECORD	1	172	H		VM LOGON AUTOLOG AND LINK RECORD
3.ACCOUNT/ NUMBER	118	8	C		VM ACCOUNT NUMBER
3.ACCT/NO	118	8	C		VM ACCOUNT NUMBER
3.TERM/ADDR	126	4	C		TERMINAL ADDRESS
3.INV/PSWD	130	8	C		INVALID PASSWORD
3.TARGET/USERID	138	8	C		MINIDISK OWNER OR AUTOLOG SUBJECT
3.DISK/ADDR	146	3	C		MINIDISK ADDRESS - CUU
3.INV/COUNT	149	2	B	SP	INVALID PASSWORD COUNT
3.INV/LIMIT	151	2	B	SP	INVALID PASSWORD LIMIT
3.DIV/CODE	153	4	C		DIVISION CODE
3.CHARGE/CODE	157	4	C		USER DEFINED CHARGE CODE
3.CHG/CODE	157	4	C		USER DEFINED CHARGE CODE
3.APPL/CODE	161	8	C		USER DEFINED APPLICATION CODE
3.PROG/CODE	169	4	C		USER DEFINED PROGRAM CODE
3.PGMR/CODE	169	4	C		USER DEFINED PROGRAM CODE

Name	Pos	Len	Format	Suffixes	Description
4..4 RECORD	1	207	H		VM USER ACCOUNTING RECORD
4.USER INFO	118	70	C		USER ACCOUNTING DATA
4.DIV/CODE	188	4	C		DIVISION CODE
4.CHARGE/CODE	192	4	C		USER DEFINED CHARGE CODE
4.CHG/CODE	192	4	C		USER DEFINED CHARGE CODE
4.APPL/CODE	196	8	C		USER DEFINED APPLICATION CODE
4.PROG/CODE	204	4	C		USER DEFINED PROGRAM CODE
4.PGMR/CODE	204	4	C		USER DEFINED PROGRAM CODE
9..9 RECORD	1	214	H		SHIFT CONTROL RECORD
9.SHIFT1	118	4	B	.4 L6 NV	SHIFT 1 START TIME
9.SHIFT2	122	4	B	.4 L6 NV	SHIFT 2 START TIME
9.SHIFT3	126	4	B	.4 L6 NV	SHIFT 3 START TIME
9.SHIFT4	130	4	B	.4 L6 NV	SHIFT 4 START TIME
9.COMPANY NAME	134	40	C		COMPANY NAME
9.PER/START/DATE	174	3	P	YYDDD	PERIOD START DATE
9.PER/START/TIME	177	4	B	.4 HM NV	
9.PER/STOP/DATE	181	3	P	YYDDD	PERIOD STOP DATE
9.PER/STOP/TIME	184	4	B	.4 HM NV	
9.NO/OF/SHIFTS	188	1	C		NO OF SHIFTS, 0 - NOT SPLIT 1-4 SHIFTS
9.FILE/IND	189	1	C		FILEIND P-PER, E-EOP, D-DLY, H-HST, A-AF
9.LONGEST/SHIFT	190	1	C		LONGEST SHIFT
9.TOTAL/SHIFT	191	4	B	.4 L6 NV	TOTAL SHIFT
9.RESERVED	195	20	C		SUMMARIZATION CONTROL INDICATOR Y OR N
8..8 RECORD	1	200	H		NETWORK ACCOUNTING INTERFACE RECORD
8.APPLICATION/ID	118	8	C		APPLICATION ID
8.CHAR/TRANS/IN	128	4	B	L7	NUMBER OF CHAR TRANSMITTED IN

Name	Pos	Len	Format	Suffixes	Description
8.CHAR/TRANS/OUT	132	4	B	L7	NUMBER OF CHAR TRANSMITTED OUT
8.RE TRANS/IN	136	4	B	L7	NUMBER OF CHAR RE-TRANSMITTED IN
8.RE TRANS/OUT	140	4	B	L7	NUMBER OF CHAR RE-TRANSMITTED OUT
8.MESS/TRANS/IN	144	4	B	L7	NUMBER OF MESSAGES TRANSMITTED IN
8.MESS/TRANS/OUT	148	4	B	L7	NUMBER OF MESSAGES TRANSMITTED OUT
8.NETWORK/ACCT	152	48	C		ACCOUNT NUMBER OF NETWORK USER
T..T RECORD	1	271	H		TAPE VOLUME ACCOUNTING RECORD
T.SLOT/NUMBER	118	4	C		TAPE SLOT NUMBER
T.CREATE/TIME	122	4	P	L7	TAPE CREATION TIME
T.CREATE/DATE	126	4	P	L7	TAPE CREATION DATE
T.SCRATCH/DATE	130	4	P	L7	VOLUME SCRATCH DATE
T.VOLUME/SERIAL	134	6	C		VOLUME SERIAL NUMBER OF TAPE
T.TECH/CODE	140	1	H		RECORD TECH CODE
T.RECORDING/ TECH	141	25	C		RECORDING TECH
T.DENSITY/CODE	166	1	H		TAPE DENSITY CODE
T.DENSITY	167	4	C		TAPE DENSITY
T.LABEL/CODE	171	1	H		TAPE LABEL TYPE CODE
T.LABEL/TYPE	172	4	C		TAPE LABEL TYPE
T.VOL/LOCATION	176	4	C		TAPE VOLUME LOCATION
T.VOL/MOVE/DATE	180	4	P	L7	TAPE VOLUME MOVE DATE
T.SYSTEM/FROM	184	1	C		TAPE SYSTEM FROM
T.DSN	185	44	C		DATA SET NAME
T.ACCOUNT/ NUMBER	229	41	C		TAPE ACCOUNT NUMBER
T.VOL/RECREATE	270	1	C		VOLUME RECREATE
Y..Y RECORD	1	154	H		z/OS SYSOUT RECORD
Y.SYSOUT/CLASS	118	1	C		SYSOUT CLASS

Name	Pos	Len	Format	Suffixes	Description
Y.SYSOUT/TYPE	119	1	C		1=STD, 2=SPECIAL, 3=PUNCH
Y.SYSOUT/OPER/ INT	120	1	C		OPERATOR INTERVENTION CODE
Y.SYSOUT/DEVICE	121	8	C		OUTPUT DEVICE NAME
Y.SYSOUT/NO/ LINES	129	6	P		SYSOUT NUMBER OF LINES PRINTED
Y.SYSOUT/PAGES	135	5	P		ESTIMATED NUMBER OF PAGES PRINTED
Y.SYSOUT/FORMSID	140	4	C		SYSOUT FORMS ID
Y.SYSOUT/RTE/CDE	144	3	C		OUTPUT ROUTE CODE
Y.RESERVED	147	8	C		
B..B RECORD	1	730	H		z/OS EXECUTION RECORD
B.RESERVE01	118	1	C		
B.FILE/LEVEL	119	1	C		FILE LEVEL
B.RCD/IND	120	1	P	NV	RECORD INDICATOR
B.RESERVE02	121	20	C		
B.STEP/JOB/IND	141	2	C		STEP/JOB INDICATOR 1B STEP RECORD
B.RESERVE05	143	2	B		
B.RESERVE06	145	1	C		
B.USER/ID	146	8	C		BLANK UNLESS FILLED BY SMF USER EXIT
B.RESERVE07	158	6	C		
B.RESERVE08	164	6	C		
B.CL	170	2	C		CLASS
B.RESERVE09	172	6	C		
B.PROGRAMMER	178	20	C		FROM OS JOB STATEMENT
B.CONV ID	180	18	C		APPC CONVERSATION ID
B.USER INFO	198	16	C		
B.ACCT FLDS	198	16	C		
B.CC	214	4	C		COMPLETION/ CANCEL CODE
B.TI	218	2	C		TERMINATION INDICATOR

Name	Pos	Len	Format	Suffixes	Description
B.STEP/NAME	220	8	C		
B.REMOTE/NAME	220	8	C		
B.PROGRAM/NAME	228	8	C		
B.ORIGINAL/TP NAME	228	8	C		APPC ORIGINAL TP NAME
B.RJE/LINE/ PASSWRD	228	8	C		
B.CORE/ALOC	236	4	P	L5 MX	1K INCREMENTS
B.CORE/USED	240	4	P	L5 MX	1K INCREMENTS
B.PCT/K	244	3	P	L3 MX	99999C; PERCENT OF UNUSED CORE
B.DISK/MOUNTS	247	5	C		
B.TSU/NBR	247	5	C		
B.P/R	252	2	C		PRIORITY
B.CARDS/READ	254	6	P	L9	
B.LINES/PRINTED	260	6	P	L10	
B.SPCL LINES/PRNTD	266	6	P	L10	
B.CARDS/PUNCHED	272	6	P	L9	
B.PRNT/FORM	278	4	C		
B.PNCH/FORM	282	4	C		
B.IDLE/TIME/HRS	286	6	P	.5 L10	999999V99999C; HOURS
B.RESERVE10	292	6	P		
B.CPU/TIME/HRS	298	6	P	.5 L10	999999V99999C; HOURS
B.OVHEAD/TIME/ HRS	304	6	P	.5 L10	999999V99999C; HOURS; OVERHEAD TIME
B.TCB/TIME	304	6	P	.5 L10	999999V99999C; HOURS; APPC TCB TIME
B.ACT/TIME/RJE/ HRS	304	6	P	.5 L10	999999V99999C; HOURS; ACTIVE RJE TIME
B.WAIT/TIME/HRS	310	6	P	.5 L10	999999V99999C; HOURS
B.CNT/TIME/RJE/ HRS	310	6	P	.5 L10	999999V99999C; HOURS; CONNECT RJE TIM
B.SRB/TIME	310	6	P	.5 L10	999999V99999C; HOURS; APPC SRB TIME

Name	Pos	Len	Format	Suffixes	Description
B.RDR Q/TIME/HMS	316	6	N	NV	READER QUEUE TIME AVERAGE (HHMMSS)
B.RDR Q/TIME/HH	316	2	N	NV	READER QUEUE TIME HOURS
B.RDR Q/TIME/MM	318	2	N	NV	READER QUEUE TIME MINUTES
B.RDR Q/TIME/SS	320	2	N	NV	READER QUEUE TIME SECONDS
B.WTR Q/TIME/HMS	322	6	N	NV	WRITER QUEUE TIME AVERAGE (HHMMSS)
B.WTR Q/TIME/HH	322	2	N	NV	WRITER QUEUE TIME HOURS
B.WTR Q/TIME/MM	324	2	N	NV	WRITER QUEUE TIME MINUTES
B.WTR Q/TIME/SS	326	2	N	NV	WRITER QUEUE TIME SECONDS
B.TAROUND/TIME/ HMS	328	6	N	NV	TURNAROUND TIME AVERAGE (HHMMSS)
B.TAROUND/TIME/ HH	328	2	N	NV	TURNAROUND TIME HOURS
B.TAROUND/TIME/ MM	330	2	N	NV	TURNAROUND TIME MINUTES
B.TAROUND/TIME/ SS	332	2	N	NV	TURNAROUND TIME SECONDS
B.READER/IO COUNT	334	7	P	L10	
B.RJE LINE/EXCP CT	334	7	P	L10	
B.BYTES/SENT	334	7	P	L10	APPC BYTES SENT
B.PRINTER/IO COUNT	341	7	P	L10	
B.RJE LINE/NAK CT	341	7	P	L10	
B.BYTES/REC	341	7	P	L10	APPC BYTES RECEIVED
B.PUNCH/IO COUNT	348	7	P	L10	
B.RJE LINE/DATA CK	348	7	P	L10	
B.NO OF/CALLS	348	7	P	L10	APPC NUMBER OF CALLS
B.TAPE/IO COUNT	355	7	P	L10	
B.RJE/INV LG	355	7	P	L10	
B.NO OF/CONV	355	7	P	L10	APPC NUMBER OF CONVERSATIONS
B.DISK/IO COUNT	362	7	P	L10	
B.RJE LINE/ERRORS	362	7	P	L10	
B.EXCP/COUNT	362	7	P	L10	APPC EXCP COUNT

Name	Pos	Len	Format	Suffixes	Description
B.OTHER/IO COUNT	369	7	P	L10	
B.RJE LINE/TME OUT	369	7	P	L10	
B.DEV/CONN/TIME	369	7	P	.3 L10	99999999V999 SECS; APPC DEV CONNECT
B.TOTAL/IO COUNT	376	7	P	L10	
B.IO INDX	383	4	P	NV	
B.S/I	387	1	C		STORAGE INDICATOR R=REAL; V=VIRTUAL
B.APPC/TYPE	387	1	C		APPC: I=INBOUND, O=OUTBOUND, T=XACTION
B.PAGE IN/COUNT	388	7	P	L10	
B.PAGE OUT/COUNT	395	7	P	L10	
B.TOTAL/PAGES	402	7	P	L10	
B.CPU/PAGE	409	4	P	L4 NV	
B.ELAP/PAGE	413	4	P	L4 NV	
B.G/1	417	1	C		GROUP CODE 1
B.G/2	418	1	C		GROUP CODE 2
B.G/3	419	1	C		GROUP CODE 3
B.P/N	420	2	C		
B.JOB/RJE/TYPE	420	2	C		JOB/RJE TYPE: A=APPC,B=BSC, S=SNA
B.INPUT/DEVICE	422	8	C		
B.LOCAL/LU NAME	422	8	C		APPC LOCAL LU NAME
B.RDR/DURATION/ HRS	430	6	P	.5 L10	999999V99999C; HOURS
B.WTR/DURATION/ HRS	436	6	P	.5 L10	999999V99999C; HOURS
B.SWAP/COUNT	442	7	P	L10	
B.SWAP/PAGES IN	449	7	P	L10	
B.SWAP/PAGES OUT	456	7	P	L10	
B.TOTAL SWAP/PAGES	463	7	P	L10	
B.NO/RD	470	2	P	L2	# OF READERS USED
B.NO/PR	472	2	P	L2	# OF PRINTERS USED

Name	Pos	Len	Format	Suffixes	Description
B.NO/PU	474	2	P	L2	# OF PUNCHES USED
B.NO/TP	476	2	P	L2	# OF TAPES USED
B.NO/DK	478	2	P	L2	# OF DISKS USED
B.NO/OT	480	2	P	L2	# OF OTHERS USED
B.NO/DU	482	2	P	L2	# OF DISKS - PRIVATE USED
B.ND	484	2	P	L2	# OF DEVICES USED
B.TSO/TPUTS	486	4	P	L7	TSO ONLY
B.TSO/TGETS	490	4	P	L7	TSO ONLY
B.ACTIVE/TIME/ HRS	494	6	P	.5 L10	999999V99999C; HOURS
B.CONNECT/TIME/ HRS	500	6	P	.5 L10	999999V99999C; HOURS
B.ALLOC/TIME/ HRS	506	6	P	.5 L10	999999V99999C; HOURS; ALLOCATION TIME
B.RES/TIME/HRS	512	6	P	.5 L10	999999V99999C; HOURS; RESIDENT TIME
B.SRB/TIME/HRS	518	6	P	.5 L10	999999V99999C; HOURS
B.TCB/TIME/HRS	524	6	P	.5 L10	999999V99999C; HOURS
B.SERVICE/UNITS	530	6	P	L10	z/OS ONLY
B.SERV/RATE	536	3	P	L5 NV	z/OS ONLY
B.CPU/IDX	539	2	P	L3 NV	z/OS ONLY
B.PAGE/SECONDS	541	6	P	L10	z/OS ONLY
B.PER/GRP	547	3	C		z/OS ONLY
B.INP/RTE	550	3	C		z/OS ONLY
B.PRT/RTE	553	3	C		z/OS ONLY
B.RQ/ID	556	2	C		JOB REQ
B.D/I	558	1	C		(1 = MONDAY 7 = SUNDAY)
B.G/4	559	1	C		
B.G/5	560	1	C		
B.G/6	561	1	C		
B.FORMS/COUNT	562	2	P	NV	
B.O/C	564	1	C		SYSOUT CLASS
B.O/T	565	1	C		SYSOUT TYPE

Name	Pos	Len	Format	Suffixes	Description
B.I/C	566	1	C		SYSOUT INTV CODE
B.OUTPUT/DVC NAME	567	8	C		
B.PARTNER/LU NAME	567	8	C		APPC PARTNER LU NAME
B.STD LINES/PRNTD	575	6	P	L10	
B.STD PAGES/PRNTD	581	5	P	L9	
B.SPCL PAGES/PRNTD	586	5	P	L9	
B.TOT PAGES/PRNTD	591	5	P	L9	
B.TAPE/MOUNTS	596	4	P	L6	
B.JOB/NBR	600	4	P	L6	
B.ABSORP/RATE	604	4	P	L6 NV	
B.USER INFO2	608	16	C		
B.USER INFO3	624	16	C		
B.USER/FIELD	640	16	C		
B.USER/COUNT	656	5	P		
B.USER/TIME/HRS	661	5	P	.5 L10	9999V99999; HOURS
B.AUDIT/FLAGS	666	6	C		SMF AUDIT FLAGS
B.PROCESS/TIME/ HRS	672	6	P	.5 L10	999999V99999C; HOURS
B.PROCESS/ CHARGE	678	6	P	.2 L11	999999999V99C; DOLLARS
B.IO/TIME/HRS	684	6	P	.5 L10	999999V99999C; HOURS
B.IO/CHARGE	690	6	P	.2 L11	999999999V99C; DOLLARS
B.UR/CHARGE	696	5	P	.2 L9	9999999V99C; DOLLARS
B.SETUP/CHARGE	701	5	P	.2 L10	9999999V99C; DOLLARS
B.TOTAL/CHARGE	706	6	P	.2 L11	999999999V99C; DOLLARS
B.C/S	712	1	C		CHARGE SUFFIX
B.ADJUSTED/RATE	713	6	P	.2 L9 NV	999999999V99C; DOLLARS
B.CONNECT/ CHARGE	719	5	P	.2 L9	9999999V99C; DOLLARS
B.RJE/CHARGE	724	5	P	.2 L9	9999999V99C; DOLLARS
B.RSRVD01	729	1	P	NV	

Appendix H: JR70CB2

Name	Pos	Len	Format	Suffixes	Description
JR70CBF/RECORD	1	2552	C		RECORD IN CHARACTER FORMAT
HEX RECORD	1	2552	H		RECORD IN HEXADECIMAL FORMAT
RCD DESCRIPTOR	1	2	B	NV	RECORD LENGTH FROM RDW
RDW	1	2	B	NV	RECORD LENGTH FROM RDW
CPU/ID	5	4	C		ID OF CPU WHERE JOB WAS EXECUTED
SMF/ID	5	4	C		SMF ID
JOB/TIME/STAMP	9	7	H		UNIQUE JOB DATE AND TIME STAMP
INTERVAL/START	9	4	P	HM NV	INTERVAL CHOPPER START TIME
INTERVAL/LENGTH	13	3	P	HM NV	INTERVAL CHOPPER INTERVAL LENGTH
RDR/DATE	9	3	P	YYDDD	READER START DATE FOR THE JOB
SES/START/DATE	9	3	P	YYDDD	SESSION START DATE FOR NETWORK USER
RDR/START/TIME	12	8	P	.4 NV HM	
SES/START/TIME	12	8	P	.4 NV HM	SESSION START TIME FOR NETWORK USER
STEP/NO	20	2	B	NT L4	RELATIVE STEP NUMBER WITHIN THE JOB
STEP/NBR	20	2	B	NT L4	NUMBER OF STEPS AT JOB SUMMARY LEVELS
SHIFT	22	1	C		SHIFT IN WHICH THE JOB WAS RUN
GREG/DATE	23	6	C	YYMMDD	STEP START DATE
DATE	23	6	C	YYMMDD	STEP START DATE
YEAR	23	2	C		STEP START YEAR
MONTH	25	2	C		STEP START MONTH
DAY	27	2	C		STEP START DAY

Name	Pos	Len	Format	Suffixes	Description
JOB NAME	29	8	C		JOB NAME FROM THE JOB STATEMENT
JOBNAME	29	8	C		JOB NAME FROM THE JOB STATEMENT
USER ID	29	8	C		VM USER IDENTITY
TERMINAL/ID	29	8	C		TERMINAL ID
START/DATE	37	3	P	YYDDD	DATE INITIATOR/USER SELECTED THIS STEP
LOGON DATE	37	3	P	YYDDD	VM LOGON DATE
SYSOUT/DATE	37	3	P	YYDDD	SYSOUT START DATE- Y RECORD
START/TIME	40	8	P	.4 NV HM	
CONV/START/TIME	40	8	P	.4 NV HM	APPC CONVERSATION START TIME
TRAN/START/TIME	40	8	P	.4 NV HM	APPC TRANSACTION START TIME
LOGON TIME	40	8	P	.4 NV HM	LOGON TIME
SYSOUT/TIME	40	8	P	.4 NV HM	SYSOUT START TIME
STOP/DATE	48	3	P	YYDDD	DATE STEP TERMINATED
LOGOFF DATE	48	3	P	YYDDD	LOGOFF DATE
STOP/TIME	51	8	P	.4 NV HM	TIME STEP TERMINATED
CONV/STOP/TIME	51	8	P	.4 NV HM	APPC CONVERSATION STOP TIME
TRAN/STOP/TIME	51	8	P	.4 NV HM	APPC TRANSACTION STOP TIME
LOGOFF TIME	51	8	P	.4 NV HM	LOGOFF TIME
ELAPS/TIME	59	8	P	.4 HM SP	TOTAL ELAPSED TIME WITHIN THE SHIFT
ELAPSED/TIME	59	8	P	.4 HM SP	TOTAL ELAPSED TIME WITHIN THE SHIFT
CONNECT/TIME	59	8	P	.4 HM SP	CONNECT TIME
ADJST/TIME	67	8	P	.4 NV HM	START TIME ADJUSTED W/IN THE SHIFT
ADJSP/TIME	75	8	P	.4 NV HM	STOP TIME ADJUSTED W/IN THE SHIFT
REC/TYPE	83	1	C		CA JASPER/JA RECORD TYPE
SUB/REC/TYPE	84	1	C		CA JASPER/JA SUB-RECORD TYPE
SYS/TYPE	85	3	C		CA JASPER GENERATED - DOS

Name	Pos	Len	Format	Suffixes	Description
CL	85	3	C		CLASS z/OS
JOB/START/TIME	88	8	P	.4 HM NV	
NET/START/TIME	88	8	P	.4 HM NV	NETWORK START TIME FOR POST-MIDNIGHT
PHYS/CPUID	96	1	C		ID OF CPU WHERE EVENT OCCURRED
REL/LEVEL	97	3	C		RELEASE AND MODIFICATION LEVEL
REL/VER	97	1	C		RELEASE VERSION
REL/REL	98	1	C		RELEASE LEVEL
REL/MOD	99	1	C		MODIFICATION LEVEL
SAVED/REC/IND	100	1	C		SAVED RECORD INDICATOR
SORT/RUN/TYPE	101	1	C		
VM/TYPE	102	1	C		
DAY/OF/WK	103	1	C		DAY OF WEEK INDICATOR
RESERVEDB	104	4	B	NT	RESERVED COMMON SECTION BINARY FIELD
VOL/RENT/TIME	104	4	B	NT	TAPE VOLUME RENT TIME
SPLIT/IND	108	1	C		SPLIT RECORD INDICATOR
RESERVEDC	109	7	C		RESERVED COMMON SECTION CHARACTER FIELD
DATA/LENGTH	116	2	B	NT	
D..D RECORD	1	133	H		DEVICE ACTIVITY RECORD
D.DEV/ADDR	118	3	C		DEVICE ADDRESS - CUU
D.DEV/CL	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
D.DEV/CLASS	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
D.DEV/TYPE	122	4	C		DEVICE TYPE E.G. 2400, 3350 ETC.
D.SIO	126	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
D.REQUESTS/SERV	126	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
D.SHIFT/LENGTH	130	4	B	.4 HM SP NT	
E..E RECORD	1	2552	H		STEP END RECORD

Name	Pos	Len	Format	Suffixes	Description
E.STEP/COMP/CODE	118	4	C		STEP COMPLETION CODE
E.STEP/TERM/CODE	122	1	C		STEP TERMINATION CODE
E.PROGRAM/NAME	123	8	C		PROGRAM NAME FROM EXEC STATEMENT
E.PART	131	2	C		PARTITION ID
E.DISK/DATA/SETS	133	2	B		NUMBER OF DISK DRIVES USED IN THE STEP
E.NO OF/DISKS	133	2	B		NUMBER OF DISK DRIVES USED IN THE STEP
E.NO OF/TAPES	135	2	B		NUMBER OF TAPE DRIVES USED IN THE STEP
E.TAPE/DATA/SETS	135	2	B		NUMBER OF TAPE DRIVES USED IN THE STEP
E.NO OF/OTHER	137	2	B		NUMBER OF OTHER DEVICES USED IN STEP
E.OTHER/DATA/ SETS	137	2	B		NUMBER OF OTHER DEVICES USED IN STEP
E.NO OF/MISC	139	2	B		NUMBER OF MISCELLANEOUS DEVICES USED
E.MISC/DATA/SETS	139	2	B		NUMBER OF NON DISK/TAPE DEVICES USED
E.DISK/SIO	141	4	B	L7 SP	NUMBER OF DISK SIO ISSUED IN THE STEP
E.DISK/EXCP	141	4	B	L7 SP	NUMBER OF DISK SIO ISSUED IN THE STEP
E.TAPE/SIO	145	4	B	L7 SP	NUMBER OF TAPE SIO ISSUED IN THE STEP
E.TAPE/EXCP	145	4	B	L7 SP	NUMBER OF TAPE SIO ISSUED IN THE STEP
E.READER/SIO	149	4	B	L7 SP	NUMBER OF RDR SIO ISSUED IN THE STEP
E.READER/EXCP	149	4	B	L7 SP	NUMBER OF RDR SIO ISSUED IN THE STEP
E.PRINTER/SIO	153	4	B	L7 SP	NUMBER OF PRT SIO ISSUED IN THE STEP

Name	Pos	Len	Format	Suffixes	Description
E.PRINTER/EXCP	153	4	B	L7 SP	NUMBER OF PRT SIO ISSUED IN THE STEP
E.PUNCH/SIO	157	4	B	L7 SP	NUMBER OF PUN SIO ISSUED IN THE STEP
E.PUNCH/EXCP	157	4	B	L7 SP	NUMBER OF PUN SIO ISSUED IN THE STEP
E.OTHER/SIO	161	4	B	L7 SP	NUMBER OF OTHER SIO ISSUED IN STEP
E.OTHER/EXCP	161	4	B	L7 SP	NUMBER OF OTHER SIO ISSUED IN STEP
E.MISC/SIO	165	4	B	L7 SP	NUMBER OF NON DISK/TAPE SIO ISSUED
E.MISC/EXCP	165	4	B	L7 SP	NUMBER OF NON DISK/TAPE SIO ISSUED
E.CPU/TIME	169	8	P	.4 MS SP	STEP CPU TIME
E.PAGE/INS	177	4	B	L7 SP	NUMBER OF PAGE-INS DURING EXECUTION
E.PAGE/OUTS	181	4	B	L7 SP	NUMBER OF PAGE-OUTS DURING EXECUTION
E.ACTIVE/TIME	185	8	P	.4 HM SP	STEP ACTIVE TIME
E.JOB/CL	193	1	C		POWER JOB CLASS
E.JOB/CLASS	193	1	C		POWER JOB CLASS
E.JOB/PRTY	194	2	B	NT	POWER JOB PRIORITY
E.JOB/FROM/USER	196	8	C		JOB FROM USER
E.RJE/IN	196	3	C		INPUT REMOTE DEVICE IN DECIMAL
E.RJE/OUT	204	3	C		OUTPUT REMOTE DEVICE IN DECIMAL
E.JOB/TO/USER	204	8	C		JOB TO USER
E.RUN/TYPE	212	1	C		RUN TYPE - P,T,C,N,R OR U
E.SUB/RUN/TYPE	213	1	C		SUB RUN TYPE
E.DOS/JOBNAME	214	8	C		DOS JOB NAME
E.DOS/DESC	222	16	C		DOS DESCRIPTION
E.DOS/ DESCRIPTION	222	16	C		DOS DESCRIPTION

Name	Pos	Len	Format	Suffixes	Description
E.PWVS DESC	238	16	C		POWER DESCRIPTION
E.PWVS/ DESCRIPTION	238	16	C		POWER DESCRIPTION
E.PWVS/ JOBNAME	254	8	C		POWER JOB NAME
E.AA/PWVS/ JOBNAME	254	8	C		FORCE DOS FORMATTING OF OUTPUT RECORD
E.PWVS/SEQNO	262	2	B	NT	POWER SEQUENCE NUMBER
E.PWVS/ABEND	264	1	C		POWER JOB ABEND INDICATOR
E.PWVS/CANCEL	265	4	C		POWER CANCEL CODE
E.STEP/FLAG	269	1	C		STEP INDICATOR S OR L
E.RESERVED/NOJ	270	1	N	SP	RESERVED NUMBER OF VSE JOBS
E.DIV/CODE	271	4	C		DIVISION CODE
E.CHARGE/CODE	275	4	C		USER DEFINED CHARGE CODE
E.CHG/CODE	275	4	C		USER DEFINED CHARGE CODE
E.APPL/CODE	279	8	C		USER DEFINED APPLICATION CODE
E.PROG/CODE	287	4	C		USER DEFINED PROGRAM CODE
E.PGMR/CODE	287	4	C		USER DEFINED PROGRAM CODE
E.SPOOL/LINES	291	4	B	SP	NUMBER OF SPOOLED PRINT LINES FOR STEP
E.SPOOL/CARDS	295	4	B	SP	NUMBER OF SPOOLED PUNCH CARDS FOR STEP
E.SPOOL/PAGES	299	2	B	SP	NUMBER OF SPOOLED PAGES FOR STEP
E.OVERHEAD	301	8	P	.4 MS SP	STEP OVERHEAD TIME
E.ALLBOUND	309	8	P	.4 MS SP	STEP ALLBOUND TIME
E.RESERVED/NOP	317	1	N		RESERVED NUMBER OF POWER JOBS
E.RESERVED/NOS	318	1	N		RESERVED NUMBER OF EXECUTION STEPS
E.HIGH/CORE	319	3	P	NT	CORE USED
E.SETUP/ITIME	322	8	P	.4 HS	SETUP-IDLE TIME IN SECONDS
E.RESERVED PACK1	330	8	P		RESERVED PACKED FIELD 1
E.RESERVED PACK2	338	8	P		RESERVED PACK FIELD 2

Name	Pos	Len	Format	Suffixes	Description
E.RESERVED PACK3	346	8	P		RESERVED PACKED FIELD 3
E.CORE/IND	354	1	C		STORAGE INDICATOR
E.TOTAL/PAGES	355	5	P	SP	TOTAL PAGE COUNT
E.TOTAL/COUNT	360	5	P	SP	TOTAL I/O COUNT
E.NO/RDR	365	2	P		NO. OF READERS USED
E.NO/PTR	367	2	P		NO. OF PRINTERS USED
E.NO/PCH	369	2	P		NO. OF PUNCHES USED
E.NO/DKP	371	2	P		NO. OF DISK-PRIVATE USED
E.NO/DEVICES	373	2	P		NO. OF DEVICES USED
E.NO/SIO/ENT	375	2	P		NO. SIO ENTRIES ON E REC
E.NODEVIC/ RESERVED	375	2	P		NO. OF DEVICES USED
E.USER/FIELD1	377	16	C		USER DATA FIELD
E.USER/FIELD2	393	5	P	SP	USER COUNT FIELD
E.USER/FIELD3	398	8	P	SP	USER TIME FIELD
E.USER/FIELD4	406	4	B		USER FIELD 4 - BINARY
E.USER/FIELD5	410	4	B		USER FIELD 5 - BINARY
E.USER/FIELD6	414	4	B		USER FIELD 6 - BINARY
E.USER/FIELD7	418	4	B		USER FIELD 7 - BINARY
E.ORIG/SEQNO	422	2	B	NT	ORIGINAL SEQUENCE NUMBER
E.EXEC/NODE	424	8	C		NAME OF EXECUTION NODE
E.FROM/NODE	432	8	C		NAME OF FROM NODE
E.FROM/USER	440	8	C		ID OF ORIGINATING USER
E.UGPSW/ RESERVED	448	1	C		WIZUGP SIO CHOPPER PROCESSED
E.RESERVEDER	449	8	C		RESERVED CHAR FIELD
E.DEACT	457	4	B	.2 MS SP	STEP DEACTIVATION TIME
E.WORK/SET	461	4	B	NT	SIZE OF STEP WORKING SET
E.OPER/ID	465	2	C		OPERATOR ID
E.CKHPT/FLAG	467	1	C		CHECKPOINT INDICATOR
E.LEAVE/FLAG	468	1	C		

Name	Pos	Len	Format	Suffixes	Description
E.NO OF/JOBS	469	4	B	SP	= 1 FOR 1ST STEP, =0 FOR ALL OTHERS
E.NO OF/DOS JOBS	469	4	B	SP	NUMBER OF DOS JOBS WITHIN PVS JOB
E.NO OF/PWVS JOBS	473	4	B	SP	NUMBER OF POWER JOBS
E.NO OF/STEPS	477	4	B	SP	NUMBER OF EXECUTION STEPS
E.RESERVEDC	481	8	C		RESERVED CHARACTER BYTES
E.RESERVED/CPU	489	8	P	MS SP	CPU TIME IN WHOLE SECONDS
E.RESERVED/OVHD	497	8	P	MS SP	OVHD TIME IN WHOLE SECONDS
E.RESERVED/WAIT	505	8	P	MS SP	WAIT TIME IN WHOLE SECONDS
E.LAST/ENTRY	505	8	P	MS SP	LAST ENTRY IN E. RECORD FOR WIZUGP
E.LASTENT/ RESERVED	505	8	P	MS SP	LAST ENTRY IN E. RECORD FOR WIZUGP
E.SIO/ENTRIES	513	2040	H	DOS	SIO ENTRIES FOR ALL DEVICES
F..F RECORD	1	224	H		ICCF ACCOUNT RECORD
F.LIBRARY	118	2	B	NT	LIBRARY ASSIGNED TO THE USER
F.NO OF/LOGON	120	4	B	SP	NUMBER OF LOGONS
F.ENTER/REQS	124	4	B	SP	NUMBER OF ENTER KEY REQUESTS
F.EXEC/REQS	128	4	B	SP	NUMBER OF EXECUTION REQUESTS
F.EXEC/UNITS	132	8	P	.4 KH SP	NUMBER OF EXECUTION UNITS USED
F.SPACE/USAGE	140	4	B	SP	TOTAL SPACE USAGE
F.ICCF/TOTAL	144	8	P	.4 KH SP	
F.INTERACT/TIME	152	8	P	.4 KH SP	
F.RESERVED/ LTIME	160	8	P	MS SP	LOGON TIME IN WHOLE SECONDS
F.RESERVED/ PTIME	168	8	P	MS SP	PART TIME IN WHOLE SECONDS
F.USER/FIELD1	176	16	C		USER DATA FIELD
F.USER/FIELD2	192	5	P	SP	USER COUNT FIELD
F.USER/FIELD3	197	8	P	SP	USER TIME FIELD
F.DIV/CODE	205	4	C		DIVISION CODE

Name	Pos	Len	Format	Suffixes	Description
F.CHARGE/CODE	209	4	C		USER DEFINED CHARGE CODE
F.CHG/CODE	209	4	C		USER DEFINED CHARGE CODE
F.APPL/CODE	213	8	C		USER DEFINED APPLICATION CODE
F.PROG/CODE	221	4	C		USER DEFINED PROGRAM CODE
F.PGMR/CODE	221	4	C		USER DEFINED PROGRAM CODE
I..I RECORD	1	571	H		IDMS ACCOUNT RECORD
I.SYSTEM/ID	118	8	C		SYSTEM ID
I.RECORD/MTYPE	126	1	C		RECORD MAINTYPE
I.RECORD/STYPE	127	1	C		RECORD SUBTYPE
I.STATS/TYPE	128	2	B	NT	STATISTICS TYPE
I.SIGNON/ID	130	32	C		SIGN ON ID -CV
I.LOGICAL/ TERMID	162	8	C		LOGICAL TERMINAL ID - CV
I.ACCT/INFO	170	16	C		USER INFORMATION FROM JOB CARD - ER
I.PWVS/SEQNO	186	2	B	NT	POWER SEQUENCE NUMBER - ER
I.PWVS/ACCT/INFO	188	14	C		POWER USER INFORMATION 1ST 14 BYTES - ER
I.TASK/CODE	202	8	C		TASK IDENTIFIER
I.TASK/NMBR	210	2	B		TASK NUMBER - NEGATIVE IF ERU
I.COND/INDS	212	3	C		CONDENSE OPTION INDICATORS
I.COND/RECS	215	4	P	SP	NUMBER OF RECORDS CONDENSED
I.PGMS/CALLED	219	8	P	SP	NUMBER OF PROGRAMS CALLED
I.PGMS/LOADED	227	8	P	SP	NUMBER OF PROGRAMS LOADED
I.TERM/READS	235	8	P	SP	NUMBER OF TERMINAL READS
I.TERM/WRITES	243	8	P	SP	NUMBER OF TERMINAL WRITES
I.TERM/ERRORS	251	8	P	SP	NUMBER OF TERMINAL I/O ERRORS
I.PAGE/REQUESTS	259	8	P	SP	NUMBER OF PAGE REQUESTS
I.TEMPSTG/GETS	267	8	P	SP	NUMBER OF TEMP STORAGE GETS
I.TEMPSTG/PUTS	275	8	P	SP	NUMBER OF TEMP STORAGE PUTS

Name	Pos	Len	Format	Suffixes	Description
I.TEMPSTG/DELS	283	8	P	SP	NUMBER OF TEMP STORAGE DELS
I.QUEUE/READS	291	8	P	SP	NUMBER OF QUEUE READS
I.QUEUE/WRITES	299	8	P	SP	NUMBER OF QUEUE WRITES
I.QUEUE/DELETES	307	8	P	SP	NUMBER OF QUEUE DELETES
I.GETTIME/ REQUESTS	315	8	P	SP	NUMBER OF GETTIME REQUESTS
I.SETTIME/ REQUESTS	323	8	P	SP	NUMBER OF SETTIME REQUESTS
I.IO/CALLS	331	8	P	SP	NUMBER OF I/O CALLS
I.STACK/HWM	339	4	B	NT	STACK HIGH WATER MARK -MAX REQUESTS/TASK
I.USER/CPU	343	8	P	SP	USER MODE CPU TIME
I.SYSTEM/CPU	351	8	P	SP	SYSTEM MODE CPU TIME
I.TASK/WAIT	359	8	P	SP	TASK WAIT TIME
I.PAGES/READ	367	8	P	SP	NUMBER OF PAGES READ
I.PAGES/WRITTEN	375	8	P	SP	NUMBER OF PAGES WRITTEN
I.PAGES/ REQUESTED	383	8	P	SP	NUMBER OF PAGES REQUESTED
I.CALC/NOFLOW	391	8	P	SP	CALCULATED RECORDS WITHOUT OVERFLOW
I.CALC/OVERFLOW	399	8	P	SP	CALCULATED RECORDS WITH OVERFLOW
I.VIA/NOFLOW	407	8	P	SP	VIA RECORDS WITHOUT OVERFLOW
I.VIA/OVERFLOW	415	8	P	SP	VIA RECORDS WITH OVERFLOW
I.RECORDS/ REQUESTD	423	8	P	SP	TOTAL RECORDS REQUESTED
I.RCRDS/ CURRENTRU	431	8	P	SP	TOTAL RECORDS CURRENT OF RU
I.FRAGMENTS/ STORED	439	8	P	SP	TOTAL FRAGMENTS STORED
I.RECORDS/ RELOCATD	447	8	P	SP	TOTAL RECORDS RELOCATED
I.LOCKS	455	8	P	SP	NUMBER OF LOCKS
I.SELECT/LOCKS	463	8	P	SP	NUMBER OF SELECT LOCKS
I.UPDATE/LOCKS	471	8	P	SP	NUMBER OF UPDATE LOCKS
I.STRG/HWM	479	4	B	NT	STORAGE HIGH WATER MARK
I.FREESTR/ REQUESTS	483	8	P	SP	FREE STORAGE REQUESTS

Name	Pos	Len	Format	Suffixes	Description
I.SYSSERV/ REQUESTS	491	8	P	SP	SYSTEM SERVICE REQUESTS
I.RESERVED/ SMCPU	499	8	P	MS SP	SYSTEM-MODE CPU TIME IN 1/60 SECONDS
I.RESERVED/ TWAIT	507	8	P	MS SP	TASK WAIT TIME IN 1/60 SECONDS
I.RESERVED/ UMCPU	515	8	P	MS SP	USER-MODE CPU TIME IN 1/60 SECONDS
I.USER/FIELD1	523	16	C		USER DATA FIELD
I.USER/FIELD2	539	5	P	SP	USER COUNT FIELD
I.USER/FIELD3	544	8	P	SP	USER TIME FIELD
I.DIV/CODE	552	4	C		DIVISION CODE
I.CHARGE/CODE	556	4	C		USER DEFINED CHARGE CODE
I.CHG/CODE	556	4	C		USER DEFINED CHARGE CODE
I.APPL/CODE	560	8	C		USER DEFINED APPLICATION CODE
I.PROG/CODE	568	4	C		USER DEFINED PROGRAM CODE
I.PGMR/CODE	568	4	C		USER DEFINED PROGRAM CODE
J..J RECORD	1	253	H		RJE SIGNFF RECORD
J.RJE/REMOTE/ID	118	3	C		REMOTE NAME-BSC
J.REMOTE NAME	118	8	C		REMOTE NAME-SNA
J.NETO/NAME	118	8	C		NODE NAME
J.LINE NAME	126	8	C		LOGICAL UNIT/LINE NAME
J.NETO/LINE	126	8	C		LINE NAME
J.PASSWORD	134	8	C		PASSWORD
J.NETO/LINEPASS	134	8	C		NUMBER OF SIO
J.EXCP/COUNT	142	4	B	SP	NUMBER OF SIO
J.SIO/COUNT	142	4	B	SP	NUMBER OF SIO
J.NETO/EXCP	142	4	B	SP	TRANSMISSION COUNT (BSC ONLY)
J.LINE/ERROR	146	4	B	SP	NUMBER OF LINE ERRORS (BSC ONLY)
J.NETO/BUFFER	146	4	B	SP	BUFFERS RECEIVED (SNA ONLY)
J.TIMEOUT	150	4	B	SP	NUMBER OF READ TEXT TIME OUTS

Name	Pos	Len	Format	Suffixes	Description
J.INV/RESP	154	4	B	SP	NUMBER OF INVALID RESPONSES
J.LINE/ADDR	158	3	C		LINE ADAPTOR ADDRESS
J.NETO/LINEADR	158	3	C		LINE ADDRESS
J.SIGNOFF	161	2	C		SIGNOFF INDICATOR
J.NETO/CANCEL	161	2	C		CANCEL CODE
J.TERM/ERROR	163	2	B	SP	NUMBER OF TERMINAL ERRORS
J.NETO/TERR	163	2	B	SP	TERMINAL ERROR NUMBER
J.RJE/ACTIVE	165	8	P	.4 HS SP	RJE ACTIVE TIME IN SECONDS
J.RJE/IDLE	173	8	P	.4 HS SP	RJE IDLE TIME IN SECONDS
J.RJE/CONNECT	181	8	P	.4 HS SP	RJE CONNECT TIME IN SECONDS
J.RJE/USER/INFO	189	16	C		USER INFO FIELD FROM SIGNON COMMAND
J.USER/FIELD1	205	16	C		USER DATA FIELD
J.USER/FIELD2	221	5	P	SP	USER COUNT FIELD
J.USER/FIELD3	226	8	P	SP	USER TIME FIELD
J.DIV/CODE	234	4	C		DIVISION CODE
J.CHARGE/CODE	238	4	C		USER DEFINED CHARGE CODE
J.CHG/CODE	238	4	C		USER DEFINED CHARGE CODE
J.APPL/CODE	242	8	C		USER DEFINED APPLICATION CODE
J.PROG/CODE	250	4	C		USER DEFINED PROGRAM CODE
J.PGMR/CODE	250	4	C		USER DEFINED PROGRAM CODE
K..K RECORD	1	259	H		
K.POWER/ID	118	1	C		POWER ID
K.NETK/JES2/JES3	118	1	C		JES2/JES3/ POWER ID
K.JOB/SFX	119	1	C		JOB SUFFIX NUMBER
K.NETK/JOB/SFX	119	1	C		JOB SUFFIX NUMBER
K.ORIG/NAME	120	8	C		NODE NAME (ORIGIN)
K.NETK/NAME	120	8	C		NODE NAME (ORIGIN)
K.EXEC/NAME	128	8	C		EXECUTION NODE NAME
K.NETK/EXEC	128	8	C		EXECUTION NODE NAME

Name	Pos	Len	Format	Suffixes	Description
K.DEST/NAME	136	8	C		DESTINATION NODE NAME
K.NETK/DEST	136	8	C		DESTINATION NODE NAME
K.JOBN/ORIG	144	2	B	NT	ORIGINAL JOB NUMBER
K.NETK/JOBN/ ORIG	144	2	B	NT	ORIGINAL JOB NUMBER
K.JOBN/CURR	146	2	B	NT	CURRENT JOB NUMBER
K.NETK/JOBN/ CURR	146	2	B	NT	CURRENT JOB NUMBER
K.ORIG/USERID	148	8	C		ORIGINAL USER IDENTIFIER
K.NETK/USER ORIG	148	8	C		ORIGINAL USER IDENTIFIER
K.NETK/DEV NAME	148	8	C		
K.USER DEST	156	8	C		DESTINATION USER ID
K.NETK/USER DEST	156	8	C		DESTINATION USER ID
K.TP COUNT	164	4	B	SP	RECORD COUNT
K.NETK/TP COUNT	164	4	B	SP	RECORD COUNT
K.CB COUNT	168	4	B	SP	COMPRESSED REC/CONTROL REC COUNT
K.NETK/CB COUNT	168	4	B	SP	COMPRESSED REC/CONTROL REC COUNT
K.USER INFO	172	16	C		USER ACCOUNTING INFORMATION
K.NETK/USER INFO	172	16	C		USER ACCOUNTING INFORMATION
K.ACCOUNT/NMBR	188	8	C		NETWORK ACCOUNT NUMBER
K.NETK/ACCOUNT	188	8	C		NETWORK ACCOUNT NUMBER
K.DEPT	196	8	C		DEPARTMENT NUMBER
K.NETK DEPT	196	8	C		DEPARTMENT NUMBER
K.BLDG	204	8	C		BUILDING NUMBER
K.NETK BLDG	204	8	C		BUILDING NUMBER
K.LOC	212	8	C		LOCATION NUMBER
K.NETK LOC	212	8	C		LOCATION NUMBER
K.ADJ/PATH	220	8	C		ADJACENT NODE NAME
K.NETK/ADJ PATH	220	8	C		ADJACENT NODE NAME
K.NETK/CANCEL	228	2	C		CANCEL CODE
K.LINADR	230	3	C		LINE ADDRESS OF SNA

Name	Pos	Len	Format	Suffixes	Description
K.NETK/LINADR	230	3	C		LINE ADDRESS OF SNA
K.PRTY	233	1	C		PRIORITY
K.NETK/PRTY	233	1	C		PRIORITY
K.CLASS	234	1	C		CLASS
K.NETK/CLASS	234	1	C		CLASS
K.QUE/ID	235	1	C		QUEUE RECORD IDENTIFIER
K.NETK/QUE/ID	235	1	C		QUEUE RECORD IDENTIFIER
K.COPIES	236	2	B		NUMBER OF COPIES
K.NETK/COPIES	236	2	B		NUMBER OF COPIES
K.JOB/TYPE	238	2	C		JOB TYPE INDICATOR
K.NETK/JOB/TYPE	238	2	C		JOB TYPE INDICATOR
K.DIV/CODE	240	4	C		DIVISION CODE
K.CHARGE/CODE	244	4	C		USER DEFINED CHARGE CODE
K.CHG/CODE	244	4	C		USER DEFINED CHARGE CODE
K.APPL/CODE	248	8	C		USER DEFINED APPLICATION CODE
K.PROG/CODE	256	4	C		USER DEFINED PROGRAM CODE
K.PGMR/CODE	256	4	C		USER DEFINED PROGRAM CODE
L..L RECORD	1	358	H		OUTPUT WRITER RECORD
L.PROGRAM/ NAME	118	8	C		EITHER PRINTER OR PUNCH
L.LINES/PRINTED	126	4	B	L7 SP	NUMBER OF LINES PRINTED
L.CARDS/ PUNCHED	130	4	B	L7 SP	NUMBER OF CARDS PUNCHED
L.PRINT/TIME	134	8	P	.4 HM SP	TIME TAKEN FOR PRINTING
L.PUNCH/TIME	142	8	P	.4 HM SP	TIME TAKEN FOR PUNCHING
L.FORM/NO	150	4	C		FORM NUMBER
L.FROM/USER	154	8	C		FROM USER
L.SYSOUT/CL	162	1	C		SYSOUT CLASS
L.SYSOUT/CLASS	162	1	C		SYSOUT CLASS
L.PAGE/COUNT	163	4	B	L6 SP	NUMBER OF PAGES PRINTED
L.OUTPUT/PRTY	167	2	B	NT	OUTPUT PRIORITY
L.JOB/PRTY	169	2	B	NT	JOB PRIORITY

Name	Pos	Len	Format	Suffixes	Description
L.JOB/FROM/USER	171	8	C		JOB FROM USER
L.RJE/IN	171	3	C		INPUT REMOTE DEVICE DECIMAL
L.JOB/TO/USER	179	8	C		JOB TO USER
L.RJE/OUT	179	3	C		OUTPUT REMOTE DEVICE IN DECIMAL
L.RUN/TYPE	187	1	C		RUN TYPE - P,T,C,N,R OR U
L.SUB/RUN/TYPE	188	1	C		SUB RUN TYPE
L.DOS/JOBNAME	189	8	C		DOS JOB NAME
L.DOS DESC	197	16	C		DOS DESCRIPTION
L.DOS/ DESCRIPTION	197	16	C		DOS DESCRIPTION
L.PWVS DESC	213	16	C		POWER DESCRIPTION
L.PWVS/ DESCRIPTION	213	16	C		POWER DESCRIPTION
L.PWVS/JOBNAME	229	8	C		POWER JOB NAME
L.AA/PWVS/ JOBNAME	229	8	C		FORCE DOS FORMATTING OF OUTPUT RECORD
L.PWVS/SEQNO	237	2	B	NT	POWER SEQUENCE NUMBER
L.PWVS/ABEND	239	1	C		POWER JOB ABEND INDICATOR
L.PWVS/CANCEL	240	4	C		POWER CANCEL CODE
L.PWVS/TRKS	244	4	B	SP	NUMBER OF TRACKS USED FOR STORAGE
L.DEV/ADDR	248	3	C		OUTPUT DEVICE ADDRESS, SNA OR GSP
L.JOB COMP CODE	251	4	C		COMPLETION CODE
L.JOB TERM CODE	255	1	C		TERMINATION CODE
L.PWVS/SUFFIX	256	2	B	NT	JOB SUFFIX NUMBER
L.COPIES	258	2	B		NUMBER OF COPIES
L.EXTRA/LINES	260	4	B	SP	NUMBER OF EXTRA LINES
L.EXTRA/PAGES	264	2	B	SP	NUMBER OF EXTRA PAGES
L.EXTRA/CARDS	266	4	B	SP	NUMBER OF EXTRA CARDS
L.PART	270	2	C		PARTITION ID
L.FLSH/ID	272	4	C		3800 FLASH IDENTIFIER

Name	Pos	Len	Format	Suffixes	Description
L.COPY/GROUP	276	8	H		COPY GROUPINGS - 3800 ONLY
L.COPY/GROUP1	276	1	B	NT	COPY GROUP FIELD 1
L.COPY/GROUP2	277	1	B	NT	COPY GROUP FIELD 2
L.COPY/GROUP3	278	1	B	NT	COPY GROUP FIELD 3
L.COPY/GROUP4	279	1	B	NT	COPY GROUP FIELD 4
L.COPY/GROUP5	280	1	B	NT	COPY GROUP FIELD 5
L.COPY/GROUP6	281	1	B	NT	COPY GROUP FIELD 6
L.COPY/GROUP7	282	1	B	NT	COPY GROUP FIELD 7
L.COPY/GROUP8	283	1	B	NT	COPY GROUP FIELD 8
L.OWN/NODE	284	8	C		NAME OF OWN NODE IN THE NETWORK
L.TO/USER	292	8	C		ORIGINATING TO NODE
L.ORIG/NODE	300	8	C		ORIGINATING FROM NODE
L.ORIG/SEQNO	308	2	B	NT	ORIGINAL JOB NUMBER
L.USER/FIELD1	310	16	C		USER DATA FIELD
L.USER/FIELD2	326	5	P	SP	USER COUNT FIELD
L.USER/FIELD3	331	8	P	SP	USER TIME FIELD
L.DIV/CODE	339	4	C		DIVISION CODE
L.CHARGE/CODE	343	4	C		USER DEFINED CHARGE CODE
L.CHG/CODE	343	4	C		USER DEFINED CHARGE CODE
L.APPL/CODE	347	8	C		USER DEFINED APPLICATION CODE
L.PROG/CODE	355	4	C		USER DEFINED PROGRAM CODE
L.PGMR/CODE	355	4	C		USER DEFINED PROGRAM CODE
M..M RECORD	1	285	H		MISCL RECORD - CREATED BY THE USER
M.ACCT INFO	118	25	C		ACCOUNTING INFORMATION
M.DIV/CODE	143	4	C		DIVISION CODE
M.CHARGE/CODE	147	4	C		USER DEFINED CHARGE CODE
M.CHG/CODE	147	4	C		USER DEFINED CHARGE CODE
M.APPL/CODE	151	8	C		USER DEFINED APPLICATION CODE
M.PGMR/CODE	159	4	C		USER DEFINED PROGRAM CODE

Name	Pos	Len	Format	Suffixes	Description
M.PROG/CODE	159	4	C		USER DEFINED PROGRAM CODE
M.MISC/CODE	163	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/COUNT	165	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/TIME	169	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/CHARGE	177	4	B	L6 SP	USER DEFINED CHARGE FOR CHARGE TYPE CODE
M.MISC/CODE1	181	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME1	183	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT1	191	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE2	195	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME2	197	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT2	205	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE3	209	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME3	211	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT3	219	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE4	223	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME4	225	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT4	233	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.MISC/CODE5	237	2	C		USER DEFINED CHARGE TYPE CODE
M.MISC/TIME5	239	8	P	.4 HM SP	USER DEFINED TIME FOR CHARGE TYPE CODE
M.MISC/COUNT5	247	4	B	L5 SP	USER DEFINED COUNT FOR CHARGE TYPE CODE
M.TOTAL/CHRG	251	6	P	.2 SP	TOTAL CHARGE
M.USER/FIELD1	257	16	C		USER DATA FIELD

Name	Pos	Len	Format	Suffixes	Description
M.USER/FIELD2	273	5	P	SP	USER COUNT FIELD
M.USER/FIELD3	278	8	P	SP	USER TIME FIELD
N..N RECORD	1	201	C		NETWORK RECORD
N.NODE/ID	118	8	C		NODE IDENTIFIER
N.NODE/PWD	126	8	C		NODE PASSWORD
N.LINE/PWD	134	8	C		LINE PASSWORD
N.INVALID/RESP	142	8	P	SP	INVALID RESPONSES PER SESSION
N.PWVS/ABEND	150	1	C		POWER JOB ABEND INDICATOR
N.PWVS/CANCEL	151	4	C		POWER CANCEL CODE
N.LINE/ADDR	155	3	C		LINE ADDRESS (CUU) OR SNA
N.TRANS/COUNT	158	8	P	SP	TRANSMISSION COUNT PER SESSION
N.TMOUT/COUNT	166	4	B	SP	TIMEOUT COUNT PER SESSION (BSC)
N.ERROR/COUNT	170	4	B	SP	ERROR COUNT PER SESSION (BSC)
N.BUFFS/RCVD	174	8	P	SP	BUFFERS RECEIVED DURING SESSION (SNA)
N.DIV/CODE	182	4	C		DIVISION CODE
N.CHARGE/CODE	186	4	C		USER DEFINED CHARGE CODE
N.CHG/CODE	186	4	C		USER DEFINED CHARGE CODE
N.APPL/CODE	190	8	C		USER DEFINED APPLICATION CODE
N.PROG/CODE	198	4	C		USER DEFINED PROGRAM CODE
N.PGMR/CODE	198	4	C		USER DEFINED PROGRAM CODE
P..O RECORD	1	406	H		INTERFACE RECORD
P.ID/FIELD1	118	2	C		ID FIELD 1
P.USER/CODE	118	2	C		UNIVERSE USERS 2-DIGIT CODE
P.ID/FIELD2	120	1	C		ID FIELD 2
P.USER/TYPE	120	1	C		UNIVERSE USER TYPE 1=DBA, 0=NONDBA
P.ID/FIELD3	121	8	C		ID FIELD 3
P.ID/FIELD4	129	16	C		ID FIELD 4

Name	Pos	Len	Format	Suffixes	Description
P.ID/FIELD41	129	4	C		ID FIELD 4 L4
P.ID/FIELD42	133	4	C		ID FIELD 4 L4
P.ID/FIELD43	137	4	C		ID FIELD 4 L4
P.ID/FIELD44	141	4	C		ID FIELD 4 L4
P.ID/FIELD5	145	16	C		ID FIELD 5
P.ID/FIELD51	145	4	C		ID FIELD 5 L4
P.ID/FIELD52	149	4	C		ID FIELD 5 L4
P.ID/FIELD53	153	4	C		ID FIELD 5 L4
P.ID/FIELD54	157	4	C		ID FIELD 5 L4
P.ID/FIELD6	161	4	C		ID FIELD 6
P.ID/FIELD7	165	4	C		ID FIELD 7
P.ID/FIELD8	169	4	C		ID FIELD 8
P.ID/FIELD9	173	2	H		ID FIELD 9
P.ID/FIELD10	175	2	B		ID FIELD 10
P.TIME/FIELD1	177	8	P	.4 MS SP	TIME FIELD 1
P.CPU/CLEAR	177	8	P	.4 MS SP	UNIVERSE CPU TIME SINCE LAST CLEAR
P.TIME/FIELD2	185	8	P	.4 MS SP	TIME FIELD 2
P.CPU/THRESH	185	8	P	.4 MS SP	UNIVERSE CPU TIME SINCE LAST THRESH.
P.TIME/FIELD3	193	8	P	.4 MS SP	TIME FIELD 3
P.TIME/FIELD4	201	8	P	.4 MS SP	TIME FIELD 4
P.TIME/FIELD5	209	8	P	.4 MS SP	TIME FIELD 5
P.TIME/FIELD6	217	8	P	.4 MS SP	TIME FIELD 6
P.TIME/FIELD7	225	8	P	.4 MS SP	TIME FIELD 7
P.TIME/FIELD8	233	8	P	.4 MS SP	TIME FIELD 8
P.TIME/FIELD9	241	8	P	.4 MS SP	TIME FIELD 9
P.TIME/FIELD10	249	8	P	.4 MS SP	TIME FIELD 10
P.COUNT/FIELD1	257	4	B	SP	COUNT FIELD 1
P.IOS/REQUESTED	257	4	B	SP	UNIVERSE IO REQS SINCE LAST CLEAR

Name	Pos	Len	Format	Suffixes	Description
P.COUNT/FIELD2	261	4	B	SP	COUNT FIELD 2
P.IOS/CLEAR	261	4	B	SP	UNIVERSE IOS SINCE LAST CLEAR
P.COUNT/FIELD3	265	4	B	SP	COUNT FIELD 3
P.IOS/THRESH	265	4	B	SP	UNIVERSE IOS SINCE LAST THRESHOLD
P.COUNT/FIELD4	269	4	B	SP	COUNT FIELD 4
P.CPU/WARNINGS	269	4	B	SP	UNIVERSE COUNT OF CPU WARNINGS
P.COUNT/FIELD5	273	4	B	SP	COUNT FIELD 5
P.IO/WARNINGS	273	4	B	SP	UNIVERSE COUNT OF IO WARNINGS
P.COUNT/FIELD6	277	4	B	SP	COUNT FIELD 6
P.RES/WAITS	277	4	B	SP	UNIVERSE NUMBER OF RESOURCE WAITS
P.COUNT/FIELD7	281	4	B	SP	COUNT FIELD 7
P.DD/COMMANDS	281	4	B	SP	UNIVERSE DATA DEF COMMANDS
P.COUNT/FIELD8	285	4	B	SP	COUNT FIELD 8
P.DD/UPDATES	285	4	B	SP	UNIVERSE DATA DEF UPDATES SINCE CLR
P.COUNT/FIELD9	289	4	B	SP	COUNT FIELD 9
P.INT/VIOLS	289	4	B	SP	UNIVERSE NUMBER OF INTEG. VIOLATIONS
P.COUNT/FIELD10	293	4	B	SP	COUNT FIELD 10
P.LUW/ROLLBKS	293	4	B	SP	UNIVERSE NUMBER OF LUW ROLLBACKS
P.COUNT/FIELD11	297	4	B	SP	COUNT FIELD 11
P.USER/DEADLKS	297	4	B	SP	UNIVERSE NUMBER OF USER DEADLOCKS
P.COUNT/FIELD12	301	4	B	SP	COUNT FIELD 12
P.USER/FORCES	301	4	B	SP	UNIVERSE NUMBER OF USER FORCES
P.COUNT/FIELD13	305	4	B	SP	COUNT FIELD 13
P.COUNT/FIELD14	309	4	B	SP	COUNT FIELD 14

Name	Pos	Len	Format	Suffixes	Description
P.COUNT/FIELD15	313	4	B	SP	COUNT FIELD 15
P.COUNT/FIELD16	317	4	B	SP	COUNT FIELD 16
P.COUNT/FIELD17	321	4	B	SP	COUNT FIELD 17
P.COUNT/FIELD18	325	4	B	SP	COUNT FIELD 18
P.COUNT/FIELD19	329	4	B	SP	COUNT FIELD 19
P.COUNT/FIELD20	333	4	B	SP	COUNT FIELD 20
P.MISC/FIELD1	337	1	H		MISC FIELD 1
P.EDITOR/PERM	337	1	H		UNIVERSE PERMISSION TO USE EDITOR
P.MISC/FIELD2	338	1	H		MISC FIELD 2
P.UGEN/PERM	338	1	H		UNIVERSE PERMISSION TO USE UGEN
P.MISC/FIELD3	339	1	H		MISC FIELD 3
P.UPREP/PERM	339	1	H		UNIVERSE PERMISSION TO USE UPREP
P.MISC/FIELD4	340	1	H		MISC FIELD 4
P.LUW/PERM	340	1	H		UNIVERSE PERMISSION TO USE LUW
P.MISC/FIELD5	341	1	H		MISC FIELD 5
P.FUNC/PERM	341	1	H		UNIVERSE PERMISSION TO USE FUNCTIONS
P.MISC/FIELD6	342	1	H		MISC FIELD 6
P.OVER/PERM	342	1	H		UNIVERSE PERM. TO OVERRIDE IO/CPU THRS
P.MISC/FIELD7	343	1	H		MISC FIELD 7
P.MISC/FIELD8	344	1	H		MISC FIELD 8
P.MISC/FIELD9	345	1	H		MISC FIELD 9
P.MISC/FIELD10	346	1	H		MISC FIELD 10
P.MISC/FIELD11	347	4	B		MISC FIELD 11
P.CPU/LIMIT	347	4	B		UNIVERSE CPU SECS BEFORE WARNING
P.MISC/FIELD12	351	4	B		MISC FIELD 12
P.IO/LIMIT	351	4	B		UNIVERSE IOS BEFORE WARNING

Name	Pos	Len	Format	Suffixes	Description
P.MISC/FIELD13	355	4	B		MISC FIELD 13
P.MISC/FIELD14	359	4	B		MISC FIELD 14
P.MISC/FIELD15	363	4	B		MISC FIELD 15
P.MISC/FIELD16	367	4	B		MISC FIELD 16
P.MISC/FIELD17	371	4	B		MISC FIELD 17
P.MISC/FIELD18	375	4	B		MISC FIELD 18
P.MISC/FIELD19	379	4	B		MISC FIELD 19
P.MISC/FIELD20	383	4	B		MISC FIELD 20
P.DIV/CODE	387	4	C		DIVISION CODE
P.CHARGE/CODE	391	4	C		USER DEFINED CHARGE CODE
P.CHG/CODE	391	4	C		USER DEFINED CHARGE CODE
P.APPL/CODE	395	8	C		USER DEFINED APPLICATION CODE
P.PROG/CODE	403	4	C		USER DEFINED PROGRAM CODE
P.PGMR/CODE	403	4	C		USER DEFINED PROGRAM CODE
R..R RECORD	1	303	H		READER RECORD
R.JOB/CL	118	1	C		POWER INPUT CLASS
R.JOB/CLASS	118	1	C		POWER INPUT CLASS
R.JOB/PRTY	119	2	B	L4 NT	POWER INPUT PRIORITY
R.JOB/FROM/USER	121	8	C		JOB FROM USER
R.RJE/IN	121	3	C		INPUT REMOTE DEVICE IN DECIMAL
R.JOB/TO/USER	129	8	C		JOB TO USER
R.RJE/OUT	129	3	C		OUTPUT REMOTE DEVICE IN DECIMAL
R.RUN/TYPE	137	1	C		RUN TYPE - P,T,C,N,R OR U
R.SUB/RUN/TYPE	138	1	C		SUB RUN TYPE
R.DOS/JOBNAME	139	8	C		DOS JOB NAME
R.DOS DESC	147	16	C		DOS DESCRIPTION
R.DOS/ DESCRIPTION	147	16	C		DOS DESCRIPTION
R.PWVS DESC	163	16	C		POWER DESCRIPTION
R.PWVS/ DESCRIPTION	163	16	C		POWER DESCRIPTION

Name	Pos	Len	Format	Suffixes	Description
R.PWVS/JOBNAME	179	8	C		POWER JOB NAME
R.AA/PWVS/ JOBNAME	179	8	C		FORCE DOS FORMATTING OF OUTPUT RECORD
R.PWVS/SEQNO	187	2	B	NT	POWER SEQUENCE NUMBER
R.PWVS/ABEND	189	1	C		POWER JOB ABEND INDICATOR
R.PWVS/CANCEL	190	4	C		POWER CANCEL CODE
R.PWVS/TRKS	194	4	B	SP	NUMBER OF TRACKS USED FOR STORAGE
R.DEV/ADDR	198	3	C		READER DEVICE ADDRESS, SNA OR PSP
R.PART	201	2	C		PARTITION ID
R.CARD/IMAGE/ RCDS	203	4	B	SP	NUMBER OF CARD IMAGE RECORDS
R.RDR/DURATION	207	8	P	.4 HS	READER DURATION TIME
R.RESERVED PACK1	215	8	P		RESERVED PACKED FIELD 1
R.RESERVED PACK2	223	8	P		RESERVED PACKED FIELD 2
R.RESERVED PACK3	231	8	P		RESERVED PACKED FIELD 3
R.OWN/NODE	239	8	C		NAME OF OWN NODE IN THE NETWORK
R.FROM/USER	247	8	C		ORIGINATING- USER ID
R.USER/FIELD1	255	16	C		USER DATA FIELD
R.USER/FIELD2	271	5	P	SP	USER COUNT FIELD
R.USER/FIELD3	276	8	P	SP	USER TIME FIELD
R.DIV/CODE	284	4	C		DIVISION CODE
R.CHARGE/CODE	288	4	C		USER DEFINED CHARGE CODE
R.CHG/CODE	288	4	C		USER DEFINED CHARGE CODE
R.APPL/CODE	292	8	C		USER DEFINED APPLICATION CODE
R.PROG/CODE	300	4	C		USER DEFINED PROGRAM CODE
R.PGMR/CODE	300	4	C		USER DEFINED PROGRAM CODE
S..S RECORD	1	163	C		SPOOL ACCESS CONNECT
S.XPCC/APPLID	118	8	C	SP	XPCC APPLICATION IDENTIFIER

Name	Pos	Len	Format	Suffixes	Description
S.NO MSGS/RETURNED	126	4	B	SP	NO. MSGS RETURNED IN RESPONSE TO CTL/PUT
S.NO CTL/REQUESTS	130	4	B	SP	NO. OF CTL REQUESTS
S.CON/TERM/CODE	134	1	H		CONNECTION- TERMINATION CODE
S.DEVICE/NAME	135	8	C		DEVICE NAME - AS DEFINED TO SUBSYSTEM
S.ACCOUNT/RECID	143	1	C		ACCOUNT- RECORD IDENTIFIER
S.DIV/CODE	144	4	C		DIVISION CODE
S.CHARGE/CODE	148	4	C		USER DEFINED CHARGE CODE
S.CHG/CODE	148	4	C		USER DEFINED CHARGE CODE
S.APPL/CODE	152	8	C		USER DEFINED APPLICATION CODE
S.PROG/CODE	160	4	C		USER DEFINED PROGRAM CODE
S.PGMR/CODE	160	4	C		USER DEFINED PROGRAM CODE
U..U RECORD	1	130	H		UNIT/DEVICE ACTIVITY RECORD
U.DEV/ADDR	118	3	C		DEVICE ADDRESS - CUU
U.DEVADDR/ RESERVED	118	3	C		DEVICE ADDRESS - CUU
U.DEV/CL	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
U.DEV/CLASS	121	1	C		DEVICE CLASS - C,D,R,T,U OR Y
U.UNIT/TYPE	122	1	P	NT	UNIT TYPE -
U.UNITTYPE	122	1	P	NT	UNIT TYPE -
U.DEV/TYPE	123	4	C		DVC TYPE-EG 3420-NOT CURRENTLY AVAILABLE
U.DEVICE/SIOS	127	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
U.REQUESTS/SERV	127	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
U.NUMSIOS/ RESERVED	127	4	B	SP	NO OF SIO REQUESTS FOR THE DEVICE
X..X RECORD	1	255	C		SPOOL ACCESS OPERATION
X.PWVS DESC	118	16	C		POWER DESCRIPTION FIELD
X.PWVS/ DESCRIPTION	118	16	C		POWER DESCRIPTION FIELD

Name	Pos	Len	Format	Suffixes	Description
X.PWVS/SEQNO	134	2	B	NT	VSE/POWER JOB SEQUENCE NUMBER
X.PWVS/CANCEL	136	4	C		POWER CANCEL CODE
X.PWVS/ABEND	140	1	C		POWER ABEND INDICATOR
X.REQ/TYPE	141	1	C		REQUEST TYPE (G=GET REQUEST P=PUT)
X.QUEUE/TYPE	142	1	C		QUEUE TYPE (R=RDR, L=LST, P=PUN)
X.JOB/SUFFIX	143	1	B	NT	JOB SUFFIX (SEGMENT) NUMBER
X.CLASS	144	1	C		CLASS OF JOB OUTPUT
X.PRTY	145	1	C		PRIORITY
X.DISP	146	1	C		DISPOSITION
X.COPIES	147	1	B	NT	NUMBER OF COPIES
X.COPY/ GROUPINGS	148	8	H		COPY GROUPINGS
X.COPY/GROUP1	148	1	B	NT	FIRST COPY GROUP FIELD
X.COPY/GROUP2	149	1	B	NT	SECOND COPY GROUP FIELD
X.COPY/GROUP3	150	1	B	NT	THIRD COPY GROUP FIELD
X.COPY/GROUP4	151	1	B	NT	FOURTH COPY GROUP FIELD
X.COPY/GROUP5	152	1	B	NT	FIFTH COPY GROUP FIELD
X.COPY/GROUP6	153	1	B	NT	SIXTH COPY GROUP FIELD
X.COPY/GROUP7	154	1	B	NT	SEVENTH COPY GROUP FIELD
X.COPY/GROUP8	155	1	B	NT	EIGHT COPY GROUP FIELD
X.PWVS/TRKS	156	2	B		NUMBER OF TRACKS FOR STORAGE
X.ORIG/SEQNO	158	2	B	NT	ORIGINAL SEQUENCE NUMBER
X.RECORDS	160	4	B		NUMBER OF RECORDS
X.EXTRA/RECS	164	4	B		NUMBER OF EXTRA RECORDS
X.LINES/CARDS	168	4	B		LINES/CARDS
X.XTRA/LINES/ CARDS	172	4	B		EXTRA LINES/CARDS
X.NO/PAGES	176	4	B		NUMBER OF PAGES
X.EXTRA/PAGES	180	4	B		NUMBER OF EXTRA PAGES

Name	Pos	Len	Format	Suffixes	Description
X.FORM/IDENT	184	8	C		FORMS ID
X.FLSH/ID	192	4	C		3800 FLASH IDENTIFIER
X.DEST/NODE	196	8	C		DESTINATION NODE
X.DEST/USER	204	8	C		DESTINATION USER
X.REQ/USER	212	8	C		REQUESTING USER ID
X.REQ/APPLID	220	8	C		REQUESTING APPLICATION ID
X.LOCAL/NODE	228	8	C		REQUESTING APPLICATION ID
X.DIV/CODE	236	4	C		DIVISION CODE
X.CHARGE/CODE	240	4	C		USER DEFINED CHARGE CODE
X.CHG/CODE	240	4	C		USER DEFINED CHARGE CODE
X.APPL/CODE	244	8	C		USER DEFINED APPLICATION CODE
X.PROG/CODE	252	4	C		USER DEFINED PROGRAM CODE
X.PGMR/CODE	252	4	C		USER DEFINED PROGRAM CODE
Z..Z RECORD	1	156	H		IPL RECORD
Z.PWR/VER	118	4	C		POWER VERSION/ MODIFICATION LEVEL
Z.PWR/LEV	122	4	C		POWER LEVEL INDICATOR
Z.PART/SIZE	126	4	B	NT	POWER PARTITION SIZE
Z.GETVS/SIZE	130	4	B	NT	POWER PARTITION GETVIS SIZE
Z.REAL/SIZE	134	4	B	NT	RESERVED PROCESSOR REAL STORAGE SIZE
Z.PWR/PART	138	2	C		POWER PARTITION ID
Z.PWR/FLAGS	140	4	C		POWER FEATURE FLAGS
Z.DFILE/EXTNTS	144	1	B	NT	NUMBER OF DATA FILE EXTENTS
Z.DFILE/TRACKS	145	4	B	NT	NUMBER OF DATA FILE TRACKS/BLOCKS
Z.QFILE/TRACKS	149	4	B	NT	NUMBER OF QUEUE FILE TRACKS/BLOCKS
Z.AFILE/TRACKS	153	4	B	NT	NUMBER OF ACCOUNT FILE TRACKS/BLOCKS
1..1 RECORD	1	185	H		VM USERID USAGE RECORD
1.ACCOUNT/ NUMBER	118	8	C		VM ACCOUNT NUMBER

Name	Pos	Len	Format	Suffixes	Description
1.ACCT/NO	118	8	C		VM ACCOUNT NUMBER
1.OVHD/CPU/TIME	126	8	P	.4 MS SP	
1.VIRT/CPU/TIME	134	8	P	.4 MS SP	
1.PAGE/READS	142	4	B	SP	NUMBER OF PAGE-READ OPERATIONS
1.PAGE/WRITES	146	4	B	SP	NUMBER OF PAGE-WRITE OPERATIONS
1.NON/SPOOL/SIO	150	4	B	SP	NUMBER OF NON-SPOOLED DEVICE SIOS
1.CARDS/ PUNCHED	154	4	B	SP	NUMBER OF VIRTUAL CARDS PUNCHED
1.LINES/PRINTED	158	4	B	SP	NUMBER OF VIRTUAL LINES PRINTED
1.CARDS/READ	162	4	B	SP	NUMBER OF VIRTUAL CARDS READ
1.DIV/CODE	166	4	C		DIVISION CODE
1.CHARGE/CODE	170	4	C		USER DEFINED CHARGE CODE
1.CHG/CODE	170	4	C		USER DEFINED CHARGE CODE
1.APPL/CODE	174	8	C		USER DEFINED APPLICATION CODE
1.PROG/CODE	182	4	C		USER DEFINED PROGRAM CODE
1.PGMR/CODE	182	4	C		USER DEFINED PROGRAM CODE
2..2 RECORD	1	156	H		VM DEDICATED DEV AND T-DISK RECORD
2.ACCOUNT/ NUMBER	118	8	C		VM ACCOUNT NUMBER
2.ACCT/NO	118	8	C		VM ACCOUNT NUMBER
2.DEV/CLASS	126	1	C		VM DEVICE CLASS CODE
2.DEV/CL	126	1	C		VM DEVICE CLASS CODE
2.DEVICE/TYPE	127	4	C		VM DEVICE TYPE
2.MODEL	131	1	H		VM DEVICE MODEL CODE
2.FEATURE	132	1	H		VM DEVICE FEATURE CODE
2.NO OF/CYLS	133	4	B	SP	NUMBER OF CYLS/SETS OF 100 BLOCKS
2.NO OF/BLOCKS	133	4	B	SP	NUMBER OF CYLS/SETS OF 100 BLOCKS

Name	Pos	Len	Format	Suffixes	Description
2.DIV/CODE	137	4	C		DIVISION CODE
2.CHARGE/CODE	141	4	C		USER DEFINED CHARGE CODE
2.CHG/CODE	141	4	C		USER DEFINED CHARGE CODE
2.APPL/CODE	145	8	C		USER DEFINED APPLICATION CODE
2.PROG/CODE	153	4	C		USER DEFINED PROGRAM CODE
2.PGMR/CODE	153	4	C		USER DEFINED PROGRAM CODE
3..3 RECORD	1	172	H		VM LOGON AUTOLOG AND LINK RECORD
3.ACCOUNT/ NUMBER	118	8	C		VM ACCOUNT NUMBER
3.ACCT/NO	118	8	C		VM ACCOUNT NUMBER
3.TERM/ADDR	126	4	C		TERMINAL ADDRESS
3.INV/PSWD	130	8	C		INVALID PASSWORD
3.TARGET/USERID	138	8	C		MINIDISK OWNER OR AUTOLOG SUBJECT
3.DISK/ADDR	146	3	C		MINIDISK ADDRESS - CUU
3.INV/COUNT	149	2	B	SP	INVALID PASSWORD COUNT
3.INV/LIMIT	151	2	B	SP	INVALID PASSWORD LIMIT
3.DIV/CODE	153	4	C		DIVISION CODE
3.CHARGE/CODE	157	4	C		USER DEFINED CHARGE CODE
3.CHG/CODE	157	4	C		USER DEFINED CHARGE CODE
3.APPL/CODE	161	8	C		USER DEFINED APPLICATION CODE
3.PROG/CODE	169	4	C		USER DEFINED PROGRAM CODE
3.PGMR/CODE	169	4	C		USER DEFINED PROGRAM CODE
4..4 RECORD	1	207	H		VM USER ACCOUNTING RECORD
4.USER INFO	118	70	C		USER ACCOUNTING DATA
4.DIV/CODE	188	4	C		DIVISION CODE
4.CHARGE/CODE	192	4	C		USER DEFINED CHARGE CODE
4.CHG/CODE	192	4	C		USER DEFINED CHARGE CODE
4.APPL/CODE	196	8	C		USER DEFINED APPLICATION CODE
4.PROG/CODE	204	4	C		USER DEFINED PROGRAM CODE
4.PGMR/CODE	204	4	C		USER DEFINED PROGRAM CODE

Name	Pos	Len	Format	Suffixes	Description
9..9 RECORD	1	214	H		SHIFT CONTROL RECORD
9.SHIFT1	118	4	B	.4 L6 NV	SHIFT 1 START TIME
9.SHIFT2	122	4	B	.4 L6 NV	SHIFT 2 START TIME
9.SHIFT3	126	4	B	.4 L6 NV	SHIFT 3 START TIME
9.SHIFT4	130	4	B	.4 L6 NV	SHIFT 4 START TIME
9.COMPANY NAME	134	40	C		COMPANY NAME
9.PER/START/DATE	174	3	P	YYDDD	PERIOD START DATE
9.PER/START/TIME	177	4	B	.4 HM NV	
9.PER/STOP/DATE	181	3	P	YYDDD	PERIOD STOP DATE
9.PER/STOP/TIME	184	4	B	.4 HM NV	
9.NO/OF/SHIFTS	188	1	C		NO OF SHIFTS, 0 - NOT SPLIT 1-4 SHIFTS
9.FILE/IND	189	1	C		FILEIND P-PER, E-EOP, D-DLY, H-HST, A-AF
9.LONGEST/SHIFT	190	1	C		LONGEST SHIFT
9.TOTAL/SHIFT	191	4	B	.4 L6 NV	TOTAL SHIFT
9.RESERVED	195	20	C		SUMMARIZATION CONTROL INDICATOR Y OR N
8..8 RECORD	1	200	H		NETWORK ACCOUNTING INTERFACE RECORD
8.APPLICATION/ID	118	8	C		APPLICATION ID
8.CHAR/TRANS/IN	128	4	B	L7	NUMBER OF CHAR TRANSMITTED IN
8.CHAR/TRANS/ OUT	132	4	B	L7	NUMBER OF CHAR TRANSMITTED OUT
8.RE TRANS/IN	136	4	B	L7	NUMBER OF CHAR RE-TRANSMITTED IN
8.RE TRANS/OUT	140	4	B	L7	NUMBER OF CHAR RE-TRANSMITTED OUT
8.MESS/TRANS/IN	144	4	B	L7	NUMBER OF MESSAGES TRANSMITTED IN
8.MESS/TRANS/OUT	148	4	B	L7	NUMBER OF MESSAGES TRANSMITTED OUT

Name	Pos	Len	Format	Suffixes	Description
8.NETWORK/ACCT	152	48	C		ACCOUNT NUMBER OF NETWORK USER
T..T RECORD	1	271	H		TAPE VOLUME ACCOUNTING RECORD
T.SLOT/NUMBER	118	4	C		TAPE SLOT NUMBER
T.CREATE/TIME	122	4	P	L7	TAPE CREATION TIME
T.CREATE/DATE	126	4	P	L7	TAPE CREATION DATE
T.SCRATCH/DATE	130	4	P	L7	VOLUME SCRATCH DATE
T.VOLUME/SERIAL	134	6	C		VOLUME SERIAL NUMBER OF TAPE
T.TECH/CODE	140	1	H		RECORD TECH CODE
T.RECORDING/ TECH	141	25	C		RECORDING TECH
T.DENSITY/CODE	166	1	H		TAPE DENSITY CODE
T.DENSITY	167	4	C		TAPE DENSITY
T.LABEL/CODE	171	1	H		TAPE LABEL TYPE CODE
T.LABEL/TYPE	172	4	C		TAPE LABEL TYPE
T.VOL/LOCATION	176	4	C		TAPE VOLUME LOCATION
T.VOL/MOVE/DATE	180	4	P	L7	TAPE VOLUME MOVE DATE
T.SYSTEM/FROM	184	1	C		TAPE SYSTEM FROM
T.DSN	185	44	C		DATA SET NAME
T.ACCOUNT/ NUMBER	229	41	C		TAPE ACCOUNT NUMBER
T.VOL/RECREATE	270	1	C		VOLUME RECREATE
Y..Y RECORD	1	154	H		z/OS SYSOUT RECORD
Y.SYSOUT/CLASS	118	1	C		SYSOUT CLASS
Y.SYSOUT/TYPE	119	1	C		1=STD, 2=SPECIAL, 3=PUNCH
Y.SYSOUT/OPER/INT	120	1	C		OPERATOR INTERVENTION CODE
Y.SYSOUT/DEVICE	121	8	C		OUTPUT DEVICE NAME
Y.SYSOUT/NO/ LINES	129	6	P		SYSOUT NUMBER OF LINES PRINTED
Y.SYSOUT/PAGES	135	5	P		ESTIMATED NUMBER OF PAGES PRINTED
Y.SYSOUT/ FORMSID	140	4	C		SYSOUT FORMS ID

Name	Pos	Len	Format	Suffixes	Description
Y.SYSOUT/RTE/ CDE	144	3	C		OUTPUT ROUTE CODE
Y.RESERVED	147	8	C		
B..B RECORD	1	730	H		z/OS EXECUTION RECORD
B.RESERVE01	118	1	C		
B.FILE/LEVEL	119	1	C		FILE LEVEL
B.RCD/IND	120	1	P	NV	RECORD INDICATOR
B.RESERVE02	121	20	C		
B.STEP/JOB/IND	141	2	C		STEP/JOB INDICATOR 1B STEP RECORD
B.RESERVE05	143	2	B		
B.RESERVE06	145	1	C		
B.USER/ID	146	8	C		BLANK UNLESS FILLED BY SMF USER EXIT
B.RESERVE07	158	6	C		
B.RESERVE08	164	6	C		
B.CL	170	2	C		CLASS
B.RESERVE09	172	6	C		
B.PROGRAMMER	178	20	C		FROM OS JOB STATEMENT
B.CONV ID	180	18	C		APPC CONVERSATION ID
B.USER INFO	198	16	C		
B.ACCT FLDS	198	16	C		
B.CC	214	4	C		COMPLETION/ CANCEL CODE
B.TI	218	2	C		TERMINATION INDICATOR
B.STEP/NAME	220	8	C		
B.REMOTE/NAME	220	8	C		
B.PROGRAM/NAME	228	8	C		
B.ORIGINAL/TP NAME	228	8	C		APPC TP ORIGINAL NAME
B.RJE/LINE/ PASSWRD	228	8	C		
B.CORE/ALOC	236	4	P	L5 MX	1K INCREMENTS
B.CORE/USED	240	4	P	L5 MX	1K INCREMENTS

Name	Pos	Len	Format	Suffixes	Description
B.PCT/K	244	3	P	L3 MX	99999C; PERCENT OF UNUSED CORE
B.JOB/NBR	247	5	C		
B.TSU/NBR	247	5	C		
B.P/R	252	2	C		PRIORITY
B.CARDS/READ	254	6	P	L9	
B.LINES/PRINTED	260	6	P	L10	
B.SPCL LINES/PRNTD	266	6	P	L10	
B.CARDS/ PUNCHED	272	6	P	L9	
B.PRNT/FORM	278	4	C		
B.PNCH/FORM	282	4	C		
B.IDLE/TIME/HRS	286	6	P	.5 L10	999999V99999C; HOURS
B.RESERVE10	292	6	P		
B.CPU/TIME/HRS	298	6	P	.5 L10	999999V99999C; HOURS
B.OVHEAD/TIME/ HRS	304	6	P	.5 L10	999999V99999C; HOURS; OVERHEAD TIME
B.TCB/TIME	304	6	P	.5 L10	999999V99999C; HOURS; APPC TCB TIME
B.ACT/TIME/RJE/ HRS	304	6	P	.5 L10	999999V99999C; HOURS; ACTIVE RJE TIME
B.WAIT/TIME/HRS	310	6	P	.5 L10	999999V99999C; HOURS
B.CNT/TIME/RJE/ HRS	310	6	P	.5 L10	999999V99999C; HOURS; CONNECT RJE TIM
B.SRB/TIME	310	6	P	.5 L10	999999V99999C; HOURS; APPC SRB TIME
B.RDR Q/TIME/HMS	316	6	N	NV	READER QUEUE TIME AVERAGE (HHMMSS)
B.RDR Q/TIME/HH	316	2	N	NV	READER QUEUE TIME HOURS
B.RDR Q/TIME/MM	318	2	N	NV	READER QUEUE TIME MINUTES
B.RDR Q/TIME/SS	320	2	N	NV	READER QUEUE TIME SECONDS
B.WTR Q/TIME/HMS	322	6	N	NV	WRITER QUEUE TIME AVERAGE (HHMMSS)

Name	Pos	Len	Format	Suffixes	Description
B.WTR Q/TIME/HH	322	2	N	NV	WRITER QUEUE TIME HOURS
B.WTR Q/TIME/MM	324	2	N	NV	WRITER QUEUE TIME MINUTES
B.WTR Q/TIME/SS	326	2	N	NV	WRITER QUEUE TIME SECONDS
B.TAROUND/TIME/ HMS	328	6	N	NV	TURNAROUND TIME AVERAGE (HHMMSS)
B.TAROUND/TIME/ HH	328	2	N	NV	TURNAROUND TIME HOURS
B.TAROUND/TIME/ MM	330	2	N	NV	TURNAROUND TIME MINUTES
B.TAROUND/TIME/ SS	332	2	N	NV	TURNAROUND TIME SECONDS
B.READER/IO COUNT	334	7	P	L10	
B.RJE LINE/EXCP CT	334	7	P	L10	
B.BYTES/SENT	334	7	P	L10	APPC BYTES SENT
B.PRINTER/IO COUNT	341	7	P	L10	
B.RJE LINE/NAK CT	341	7	P	L10	
B.BYTES/REC	341	7	P	L10	APPC BYTES RECEIVED
B.PUNCH/IO COUNT	348	7	P	L10	
B.RJE LINE/DATA CK	348	7	P	L10	
B.NO OF/CALLS	348	7	P	L10	APPC NUMBER OF CALLS
B.TAPE/IO COUNT	355	7	P	L10	
B.RJE/INV LG	355	7	P	L10	
B.NO OF/CONV	355	7	P	L10	APPC NUMBER OF CONVERSATIONS
B.DISK/IO COUNT	362	7	P	L10	
B.RJE LINE/ERRORS	362	7	P	L10	
B.EXCP/COUNT	362	7	P	L10	APPC EXCP COUNT
B.OTHER/IO COUNT	369	7	P	L10	
B.RJE LINE/TME OUT	369	7	P	L10	
B.DEV/CONN/TIME	369	7	P	.3 L10	9999999V999 SECS; APPC DEV CONNECT
B.TOTAL/IO COUNT	376	7	P	L10	
B.IO INDX	383	4	P	NV	

Name	Pos	Len	Format	Suffixes	Description
B.S/I	387	1	C		STORAGE INDICATOR R=REAL; V=VIRTUAL
B.APPC/TYPE	387	1	C		APPC: I=INBOUND, O=OUTBOUND, T=XACTION
B.PAGE IN/COUNT	388	7	P	L10	
B.PAGE OUT/COUNT	395	7	P	L10	
B.TOTAL/PAGES	402	7	P	L10	
B.CPU/PAGE	409	4	P	L4 NV	
B.ELAP/PAGE	413	4	P	L4 NV	
B.G/1	417	1	C		GROUP CODE 1
B.G/2	418	1	C		GROUP CODE 2
B.G/3	419	1	C		GROUP CODE 3
B.P/N	420	2	C		
B.JOB/RJE/TYPE	420	2	C		JOB/RJE TYPE: A=APPC,B=BSC, S=SNA
B.INPUT/DEVICE	422	8	C		
B.LOCAL/LU NAME	422	8	C		APPC LOCAL LU NAME
B.RDR/DURATION/ HRS	430	6	P	.5 L10	999999V99999C; HOURS
B.WTR/DURATION/ HRS	436	6	P	.5 L10	999999V99999C; HOURS
B.SWAP/COUNT	442	7	P	L10	
B.SWAP/PAGES IN	449	7	P	L10	
B.SWAP/PAGES OUT	456	7	P	L10	
B.TOTAL SWAP/PAGES	463	7	P	L10	
B.NO/RD	470	2	P	L2	# OF READERS USED
B.NO/PR	472	2	P	L2	# OF PRINTERS USED
B.NO/PU	474	2	P	L2	# OF PUNCHES USED
B.NO/TP	476	2	P	L2	# OF TAPES USED
B.NO/DK	478	2	P	L2	# OF DISKS USED
B.NO/OT	480	2	P	L2	# OF OTHERS USED
B.NO/DU	482	2	P	L2	# OF DISKS - PRIVATE USED

Name	Pos	Len	Format	Suffixes	Description
B.ND	484	2	P	L2	# OF DEVICES USED
B.TSO/TPUTS	486	4	P	L7	TSO ONLY
B.TSO/TGETS	490	4	P	L7	TSO ONLY
B.ACTIVE/TIME/ HRS	494	6	P	.5 L10	999999V99999C; HOURS
B.CONNECT/TIME/ HRS	500	6	P	.5 L10	999999V99999C; HOURS
B.ALLOC/TIME/ HRS	506	6	P	.5 L10	999999V99999C; HOURS; ALLOCATION TIME
B.RES/TIME/HRS	512	6	P	.5 L10	999999V99999C; HOURS; RESIDENT TIME
B.SRB/TIME/HRS	518	6	P	.5 L10	999999V99999C; HOURS
B.TCB/TIME/HRS	524	6	P	.5 L10	999999V99999C; HOURS
B.SERVICE/UNITS	530	6	P	L10	z/OS ONLY
B.SERV/RATE	536	3	P	L5 NV	z/OS ONLY
B.CPU/IDX	539	2	P	L3 NV	z/OS ONLY
B.PAGE/SECONDS	541	6	P	L10	z/OS ONLY
B.PER/GRP	547	3	C		z/OS ONLY
B.INP/RTE	550	3	C		z/OS ONLY
B.PRT/RTE	553	3	C		z/OS ONLY
B.RQ/ID	556	2	C		JOB REQ
B.D/I	558	1	C		(1 = MONDAY 7 = SUNDAY)
B.G/4	559	1	C		
B.G/5	560	1	C		
B.G/6	561	1	C		
B.FORMS/COUNT	562	2	P	NV	
B.O/C	564	1	C		SYSOUT CLASS
B.O/T	565	1	C		SYSOUT TYPE
B.I/C	566	1	C		SYSOUT INTV CODE
B.OUTPUT/DVC NAME	567	8	C		
B.PARTNER/LU NAME	567	8	C		APPC PARTNER LU NAME
B.STD LINES/PRNTD	575	6	P	L10	

Name	Pos	Len	Format	Suffixes	Description
B.STD PAGES/PRNTD	581	5	P	L9	
B.SPCL PAGES/PRNTD	586	5	P	L9	
B.TOT PAGES/PRNTD	591	5	P	L9	
B.TAPE/MOUNTS	596	4	P	L6	
B.DISK/MOUNTS	600	4	P	L6	
B.ABSORP/RATE	604	4	P	L6 NV	
B.USER INFO2	608	16	C		
B.USER INFO3	624	16	C		
B.USER/FIELD	640	16	C		
B.USER/COUNT	656	5	P		
B.USER/TIME/HRS	661	5	P	.5 L10	9999V99999; HOURS
B.AUDIT/FLAGS	666	6	C		SMF AUDIT FLAGS
B.PROCESS/TIME/ HRS	672	6	P	.5 L10	999999V99999C; HOURS
B.PROCESS/CHARGE	678	6	P	.2 L11	999999999V99C; DOLLARS
B.IO/TIME/HRS	684	6	P	.5 L10	999999V99999C; HOURS
B.IO/CHARGE	690	6	P	.2 L11	999999999V99C; DOLLARS
B.UR/CHARGE	696	5	P	.2 L9	9999999V99C; DOLLARS
B.SETUP/CHARGE	701	5	P	.2 L10	9999999V99C; DOLLARS
B.TOTAL/CHARGE	706	6	P	.2 L11	999999999V99C; DOLLARS
B.C/S	712	1	C		CHARGE SUFFIX
B.ADJUSTED/RATE	713	6	P	.2 L9 NV	999999999V99C; DOLLARS
B.CONNECT/ CHARGE	719	5	P	.2 L9	9999999V99C; DOLLARS
B.RJE/CHARGE	724	5	P	.2 L9	9999999V99C; DOLLARS
B.RSRVD01	729	1	P	NV	

Appendix I: CA JARS Wizard z/OS Copybooks

CA JARS and CA JARS DSA Option copybooks are described in this chapter.

This section contains the following topics:

CA JARS Copybooks

The copybooks provided with CA JARS for use in a CA JARS Wizard run all create time or date work fields from other CAWGJR70 glossary items. Each copybook is listed in the table below along with the work field it creates and the CAWGJR70 glossary items used to create it.

Copybook	Work Field Created	Uses CAWGJR71 Glossary Items	New Work Field Format
JRDRDATE	RDR STRT/DTE	RDR STRT/DTE/YY RDR STRT/DTE/MM RDR STRT/DTE/DD	YY/MM/DD
JRDRSTME	RDR STRT/TME	RDR STRT/TME/HH RDR STRT/TME/MM RDR STRT/TME/SS	HH:MM:SS
JRUNDATE	RUN/DATE	RUN/DATE/YY RUN/DATE/MM RUN/DATE/DD	YY/MM/DD
JSTOPTME	STOP/TIME	STOP/TIME/HH STOP/TIME/MM STOP/TIME/SS	HH:MM:SS
JSTRTIME	START/TIME	START/TIME/HH START/TIME/MM START/TIME/SS	HH:MM:SS

If you want to use these work fields in any CA JARS Wizard run, you must include a COPY statement for the appropriate CA JARS copybook.

CA JARS DSA Option Copybooks

The copybooks provided with CA JARS DSA Option for use in a CA JARS Wizard run is listed in the table below. You cannot use this copybook unless CA JARS DSA Option is installed at your site.

Copybook	Function
CAIJAWCV	Converts several CA JARS DSA Option glossary fields to appropriate output formats.

CAIJAWCV Copybook

This copybook converts the following CA JARS DSA Option glossary fields to appropriate output formats.

Note: These fields are only found in the CA JARS DSA Option master file.

Glossary Item Name	Format	Output Item Name	Format
CR/DATE	4 P	CDATE	8 C
L R/DATE	4 P	LDATE	8 C
DSORG	2 H	CDSORG	2 C
RECFM	1 H	CRECFM	3 C
UNIT/TYPE	1 H	CUNIT	6 C
LRECL	2 B	CLREC	5 C
BLKSIZE	2 B	CBLKSIZ	5 C

To obtain the converted output formats, simply reference the output item name instead of the glossary item name in your CA JARS Wizard control statements. If you use COPY CAIJAWCV in a CA JARS Wizard run, you must treat the following work fields as reserved words.These work fields are created and used by the copybook.

Work Field Name

- - - - - - - - - - - - - - -

FILL1

FILL2

FILL3

FILL4

FILL5

HDSORG

NLREC

NBLKSIZ

In addition, the following COMPUTE labels are also reserved:

COMPUTE Label Name

- - - - - - - - - - - - - - - - - -

LAB1

LAB2

LAB3

Do not use these labels or work fields in any reports where you use a COPY CAIJAWCV statement.

Index