CA SYSVIEW® Performance Management

Using the CA EXPLORE Report Writer

Release 13.5

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.

Copyright © 2012 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA SYSVIEW® Performance Management (CA SYSVIEW)
- CA SYSVIEW® Performance Management Option for CICS (CA SYSVIEW Option for CICS)

Contact CA Technologies

Contact CA Support

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

- 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.

If you would like 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

Overview 13
What Is History Reporting 13
Memory Requirements 14
Report Writer Concepts and Terminology 14
Syntax Conventions 16
    Apply Syntax Conventions 18
    Command Syntax Rules 19

Chapter 2: Planning Your Reports 21

How to Plan a Report 21
About Sample Reports 22
Types of Report Formats 22
Types of Information to Include 23
    Variables 23
    Variable Options 23
    ID Options and Identifiers 24
How to Further Tailor Information 24

Chapter 3: Writing a Report 25

How to Write Reports 25
Types of Reporting Commands 27
    Global Commands 27
    Report Commands 28
Write A Report 31
Write Multiple Reports 32
Tailor Reports Using Secondary Commands 33
    Create Multiple Tailored Reports 33
Reports with a File Type of CSV 34
    CSVGEN Utility—Convert Generated Reports to CSV Format 35
    How to Download CSV Formatted Reports 37

Chapter 4: Commands 39

About the Commands 40
Variables with Commands 40

ID Options with Commands and Variables 40
CARDLEN Command—Set the Character Length 41
COMMENT Command—Add a Comment Line 41
COST Command—Provides Charge-back Statistics 42
COUNT Command—Define Number of Records 44
DEFINE Command—Define ID Options and Variables 45
DEFINE XID, XVAR—Define New ID Options or Variables 46
DEFINE ID—Define Options Using Existing ID Options 48
DEFINE VAR—Define Variables Using Existing Variables 50
Define XID Options Using the List Command 52
EACH Command—Define a Time Period 54
END Command—End a Report 56
EXCLUDE Command—Define Excluded Time Ranges 57
EXITRTN Command—Define User Exits 58
FLASHBACK AUDITLOG Command—Generate Audit Report 59
FLASHBACK LIST Command—Generate Transaction Report 60
FROM Command—Begin Time for Report 62
How FROM and TO Relate to the Log and Historical Files 63
Actual Date and Time 63
Relative Date and Time 64
GLOBAL Command—Assign Parameter Values 66
Temporarily Override a Global Value 66
Remove a Global Value 67
GOTO Command—Jump to Specified Location 67
GRAF Command—Produce a Graph 68
GRAF2 Command—Produce Subgraphs 70
GROUP Command—Combine Resources 72
HEADER Command—Specify a Title 74
HPLOT Command—Produce a Horizontal Plot 76
IF Command—Assign Identifier Values 77
INCLUDE Command—Define Time Ranges 78
INPUT Command—Input File Name 80
INVOICE Command—Produce a Tabular Invoice Report 80
LINECOUNT Command—Set Lines Per Page 82
MPLOT Command—Produce a Vertical Plot 83
OPTION Command—Define Report Options 84
PERIOD Command—Specify a Time 91
PLOT Command—Plot Resource Usage 92
PLOT2 Command—Plot Two Variables 94
PRTEXIT Command—Define a User Exit 94
RANGE Command—Set Criteria to Include 95
RUN Command—Begin Processing and Printing Reports 97

SELECT Command—Specify Identifiers 97
SHIFT Command—Hours Covered for a Report 98
TABulate Command—Produce Tabular Reports 100
TAB2 Command—Build a Secondary Table 103
TITLE1, TITLE2, and TITLE3 Commands 104
VPLOT Command—Plot Usage in Vertical Columns 106
VPLOT2 Command—Plot Two Variables 107

Chapter 5: Variables 109

Functions of Variables and Options 109
Structure and Parts of Variable Names 110
Conventions Used for Describing Variables 111
Source for MVS Variable Values 112
JOB and STEP Record Type Data 112
SMF dsect Field Names Used When Specifying Variables 112
CICS Variables 112
CICS CDSA Variable Class 113
CICS DATASET Variable Class 113
CICS DB2 Variable Class 114
CICS DLI Variable Class 114
CICS ECDSA Variable Class 116
CICS ERDSA Variable Class 116
CICS EUDSA Variable Class 117
CICS EXCEPTION Variable Class 117
CICS PAGE Variable Class 118
CICS PROGRAM Variable Class 119
CICS REGION Variable Class 119
CICS SYSTEM Variable Class 120
CICS TEMPSTOR (Temporary Storage) Variable Class 120
CICS TERMINAL Variable Class 121
CICS THRESHOLD Variable Class 122
CICS Interval Record TRANsaction Variable Class 124
CICS Performance Record TRANsaction Variable Class 124
CICS TRANSIENT Variable Class 128
CICS UDSA Variable Class 129
IMS Variables 129
IMS TRANsaction Variable Class 129
IMS REGION Variable Class 130
IMS DLI Variable Class 131
MQ Variables 132
MQ OBJEct Variable Class 132

MQ OBJFailure Variable Class 133
MQ OBJMessage Variable Class 133
MQ OBJTimer Variable Class 134
MQ JOBStat Variable Class 135
MQ JOBFailure Variable Class 135
MQ JOBMessage Variable Class 136
MQ JOBTimer Variable Class 137
AUDIT Variables 137
AUDIT ALL Variable Class 138
COMMON Variables 138
COMMON STATE Variable Class 138
MVS Variables 140
MVS ASCB Variable Class 140
MVS CHANNEL Variable Class 141
MVS CPU Variable Class 142
MVS DEVICE Variable Class 143
MVS DISK Variable Class 143
MVS DSN Variable Class 144
MVS EXCP Variable Class 145
MVS IO Variable Class 146
MVS JOB and STEP Variable Classes 147
MVS LPAR Variable Class 148
MVS OPERator Variable Class 148
MVS PAGE Variable Class 149
MVS PAGESWAP Variable Class 151
MVS PERFormance Variable Class 151
MVS PRINT Variable Class 152
MVS PROCessor Variable Class 153
MVS STORage Variable Class 154
MVS SU Variable Class 155
MVS SWAP Variable Class 155
MVS TSO Variable Class 156

Chapter 6: Variable Options 157

Description and Syntax 157
COUNT Option 158
FORMAT Option 158
LOGICAL Option 159
MAX and MIN Options 159
MAXDATE and MAXTIME Options 160
PerCenT Option 161

RATE Option 161
SELZERO Option 163
TOTAL Option 163
XTIME Option 164
XTND and NXTND Options 165
XTND Option 165
NXTND Option 165

Chapter 7: ID Options 167

How to Use ID Options 167
Descriptions of ID Options 168
ID Options for CICS 168
ID Options for IMS 170
ID Options for MQ 171
ID Options for AUDITLOG 172
ID Options for COMMON 172
ID Options for MVS 174
ID Options Used with Commands 177
Limitations Caused by Using Identifiers 177
ID Options Syntax 178
Examples of ID Options Used with Commands 178
Use ID Options with Variables and Identifiers 181
ID Options Used with Commands or Variables 182
Sample Commands for Report 1 182
Sample Commands and Variables for Report 2 183
Generate a Plot Report Using Variables 184
Generate a Tab Report Using Commands and Variables 185
Identifiers 186
Types of Identifiers 186
Generic Characters in Identifiers 186
Specify Identifiers in Hexadecimal 187
How Format Identifiers Work 187
Limit the Number of Characters Printed 188
Define New or Redefine Existing Identifiers 189

Chapter 8: Tabular Reports 191

About Tabular Reports 191
TAB Command 192
Reports Organized by Time and Date or by Resources 192
Reports Organized by Date and Time Using Tab Operands 192
Organize by Date, Time, or Both and Resource 196

Specify Variables 196
ID Options Specified with the TAB Command 196
Include CICS and MVS Variables 197
Use ID Options with Variables 198
Variable Options 200
DIFF Option—Percentage of Change Between Time Intervals 201
RANGE Option—Test Range Criteria 201
SELECT Option—Specify a Time Period for a Variable 202
RIGHT Option—Shift the Column for a Variable to the Right 203
RATE OPTION—Display a Rate Instead of a Count 204
Limit the Contents of a Report 204
Limit the Resources Included in a Report 205
FOR Operand—Limit the Identifiers Included in a Report 205
FOR Operand—Limit Reports Organized by Time and Date 206
TOP and BOTTOM Operands—Limit Reports 207
How Report Sorting Works 209
Sort Reports by Different Keys 209
SORT ASCENDING Operand—Sort in Ascending Order 210
Sort Reports by Identifier 211
Use Other Commands with the TAB Command 212
How the RANGE Command Specifies Ranges 212
Define Groups of Resources Using the Group Command 215
Define Groups by Values Within Specified Ranges 216
Use the COST Command to Create Charge-Back Reports 218
Create Charge-Back Reports Organized by Group 219
TAB2 Command 220
Limit TAB2 Report Use the FOR Operand 220
Limit TAB2 Reports Using the TOP or BOTTOM Operand 222

Chapter 9: Plot Reports 225

What You Need to Know 225
PLOT Command—Create a Simple PLOT Report 226
Plot the Variable Value for a Specified Resource 227
ALONE Operand—Exclude Global Values from a Plot 228
SCALE Operand—Change the Scale of a PLOT Report 229
CHARn Operands—Specify Plot Characters 230
RATE Option—Convert Counts to Rates 231
PLOT2 Command—Plot Response Time 231
PLOT2 RATE Command—Convert Counts to Rates 233
VPLOT Command—Create a Simple Report 234
SCALE(3X) Operands—Widen the Scale of a VPLOT Report 235

SCALE (n) Operand—Specify the Maximum Scale Value 236
STACK=id Operand—Replace the Plot Character 237
VPLOT2 Command—Create a Simple VPLOT2 Report 238
SCALE(COMMON) Operand—Use the Same Scale 238
MPLOT Command—Used with One and Multiple Variables 239
HPLOT Command 241
SCALE Operand—Used with HPLOT and VPLOT 241
EACH RECORD Command—Used with VPLOT and HPLOT 243
EACH RECORD Command—Used with MPLOT and HPLOT 244
SCALE Command—Change the Scale 246

Chapter 10: Graph Reports 247

GRAF Command—Create a GRAF Report 247
GRAF JOBNAME—Graph by an Alternate Resource 248
TRANID ID Option—Specify the Resources to be Included in a Graph 248
FOR Operand—Limit Lines in a Graph 249
SCALE(n) Operand—Change the Scale of a Graph 250
SORT ASCENDING—Sort Graph Reports 251
SHIFT Command—Create Reports on Specified Shifts 252
EACH Command—Create Reports with Multiple Graphs 253
RANGE Command—Create Reports with a Specified Range 254
GRAF2 Command Restrictions 255
FOR Operand—Limit the Number of Identifiers in a GRAF2 Report 256
SCALE Operand—Change the Scale of a GRAF2 Report 257
GRAF2 TRANNUM—GRAF2 Command with an Identifier 257
GRAF2 Command Used with Multiple Variables 258

Chapter 11: Historical Reports 261

About Historical Reports 261
Output Considerations 262
FLASHBACK LIST Command 262
Default CICS FLASHBACK LIST Report 262
Create a Default FLASHBACK IMS Report 265
Create a Default FLASHBACK AUDITLOG Report 266
Limit FLASHBACK Reports to Specified Resources 267
Create Report with Specified Columns 269
Create a Customized FLASHBACK LIST Report 270
Limit Default FLASHBACK Reports to Specified Variable Values 271

Chapter 12: Messages 273

Syntax Error Messages 275
Other Messages 275
ERPT280E 297

Appendix A: User Exits 301

About User Exits 301
Supplied User Exits 301
Source Code 302

Appendix B: Sample Reports 303

List of Sample Reports 303
CICS Sample Reports 303
IMS Sample Reports 307
MQ Sample Reports 308
MVS Sample Reports 309

Index 311

Chapter 1: Introduction

This section contains the following topics:

Overview (see page 13)
What Is History Reporting (see page 13)
Memory Requirements (see page 14)
Report Writer Concepts and Terminology (see page 14)
Syntax Conventions (see page 16)

Overview

This CA SYSVIEW guide explains how to use the history reporting commands and variables to create reports.

Requirements for the person writing reports:

- Functional knowledge of z/OS, CICS, AUDIT, IMS, and WebSphere MQ
- Ability to edit a file and access SMF records

What Is History Reporting

The CA SYSVIEW history reporting subsystem, CA EXPLORE Report Writer, lets you produce z/OS, CICS, AUDIT, IMS, and WebSphere MQ graphic and tabular reports. The reports are produced from the performance data collected by CA SYSVIEW data collectors. With these reports, you can study long-term resource usage or detect and analyze unusual events.

The Report Writer reports on many CICS activities, such as transaction use, maximum terminal response time through the CA SYSVIEW Option for CICS. You control the time that each detail line in a report represents and the time frame that a report covers.

The Report Writer lets you access the information you want, at the level of detail you indicate. With one pass through the historical data, you can create up to 50 reports, each covering a different time period.

Memory Requirements

The amount of virtual storage needed by the Report Writer depends on the following:

- Types of reports you want to produce
- Number of reports requested in each run
- Amount of input data used

To reduce the amount of virtual storage required:

- Limit the number of reports in each run
- Narrow the time period covered by each report

We recommend that you run the Report Writer in a region of at least 4 MB.

Report Writer Concepts and Terminology

This section describes the major concepts and terminology that enable you to use the Report Writer more efficiently.

Report Writer commands

Used with JCL to produce reports. Report Writer commands include global commands and report commands.

Examples: INPUT, RUN, PLOT, EACH, TITLE1

Types of Report Writer commands include the following:

Global commands

Affects all reports in a run.

Examples: COUNT, INPUT, RUN

Report commands

Affects only a single report. Report commands include primary report commands and secondary report commands.

Examples: PLOT, EACH, TITLE1

Types of Report commands include the following:

Primary report commands

Defines the type of report to generated.

Examples: PLOT, TAB2

Secondary report commands

Alters or enhance a report.

Examples: EACH, FROM, TITLE1

The Report Writer commands can contain the following parts:

Operands

Keywords specified with commands or variable options that affect the content or structure of a report. Different commands and variable options have different operands.

Example: Partial INCLUDE syntax

```
INCLUDE [day]
INCLUDE SATURDAY
```

Saturday is the operand.

Variables

Structured keywords that specify what the report is about.

Examples: The following keywords include transaction usage and terminal response.

TRAN USE, TERM RESP

Variable options

Keywords that change either the type of data a variable represents or the format of that data.

Examples: AVeraGe, FORMAT, RATE

ID options

Keywords that change either the scope of the data that a variable represents or organize the data in a tabular or flashback report by a type of resource.

Examples: TRANID, TERMID, USERID

Identifiers

Either the name of a particular resource or a generic specification of a group of resources that a report or variable is limited to. Identifiers are specified following ID options.

Examples: MYTRAN, ¬OPER1, TERM1+, CICS*

Syntax Conventions

Command syntax is shown using the following conventions:

- **Uppercase letters-**Uppercase letters indicate that you must enter the entire word.

  Sample Syntax:

  ```
  PLOT
  ```

  Sample Entry:

  ```
  PLOT
  ```

- **Lowercase italics letters-**Variables are presented in lowercase italics letters. Substitute specific information for each variable.

  Sample Syntax:

  ```
  PLOT variable idoption(identifier)
  ```

  Sample Syntax:

  ```
  PLOT TRAN USE TRANID(CINQ)
  ```

  In JCL, italics indicate user-supplied information. For example, the following line appears in the JCL for creating reports using tape log file data.

  ```
  INPUT(filename)
  ```

  When *filename* is italicized, you do not type *filename.* Instead, replace it with the filename of your tape log file, as explained in the description of the JCL.

- **Mixed case-**Mixed case indicates you can enter either the uppercase portion or the entire word.

  Sample Syntax:

  ```
  PerCenT
  ```

  Sample Entry:

  ```
  PCT
  ```

- **{ }-**Braces enclose options that you are required to enter. Enter exactly one of the items listed vertically. Enter one or more of the items listed horizontally.

  Sample Syntax:

  ```
  SCALE({mX},{n},{COMMON})
  ```

  Sample Entry:

  ```
  SCALE(3X,COMMON)
  ```

- **[ ]**-Brackets enclose options that you are not required to enter. You can enter one or none of the options listed vertically. You can enter more than one of the options listed horizontally, in the order indicated.

  Sample Syntax:

  ```
  [HDR1(text)] [HDR2(text)] [FOR n]
  [TOP n] [BOTTOM n]
  ```

  Sample Entry:

  ```
  HDR2(Rate) TOP 5
  ```

- **( )**-Parentheses indicate that you must enter parentheses around the item as shown.

  Sample Syntax:

  ```
  [SCALE({mX},{n},{COMMON})]
  ```

  Sample Entry:

  ```
  SCALE(3X)
  ```

- **|**-An or-sign indicates a choice between related items. Enter only one of the items.

  Sample Syntax:

  ```
  PGMSTOR=NO|YES
  ```

  Sample Entry:

  ```
  PGMSTOR=NO
  ```

- **__**-Underscoring indicates the default. If you want only the underscored parameters of a command, you do not have to specify the command.

  Sample Syntax:

  ```
  OPTION {(ECHO=YES) }{(ECHO=NO) }
  ```

  Sample Entry:

  You need no command to specify:

  ```
  OPTION(ECHO=YES)
  ```

- **...**-An ellipsis indicates that the preceding item can be continued.

  Sample Syntax:

  ```
  TAB variable [AND,...]
  ```

Apply Syntax Conventions

To help you apply command syntax in context, this section explains valid and invalid ways to enter the PLOT command.

The PLOT command has the following format:

```
PLOT variable [options] [ALONE] [SCALE({n},{mX})] [CHAR1(x)]
                                                  [CHAR2(x)]
```

The following table shows invalid entries of the PLOT command, explains why they are invalid, and shows a sample corrected entry:

Invalid Entry Segment	Reason the Entry Is Invalid	Corrected Entry Segment
PLOT *variable*	The word *variable* is in lowercase italic letters. Replace *variable* with a particular variable name.	PLOT TRAN USE
PLOT TRAN USE SCALE	*m*X and *n* are listed horizontally in braces, indicating that you must include at least one of them when you use the SCALE operand.	PLOT TRAN USE SCALE(50)
PLOT TRAN USE CHAR1 *	*x* is in parentheses. Include the parentheses with each parenthetical option you specify.	PLOT TRAN USE CHAR1(*)

Command Syntax Rules

History reporting commands must follow certain syntax rules.

When the Report Writer encounters a syntax error, it does the following:

- Displays a dollar sign ($) underneath the unexpected word or character
- Includes a message about a possible cause

When the Report Writer encounters a syntax error, it does the following:

- Continues to read commands to look for more syntax errors
- Does not generate a report

Keep in mind the following points when writing history reporting commands:

- Commands and their operands can be entered in either uppercase or lowercase letters.
- Each command must go on a new line. Do not enter more than one command on a line.
- Commands must be the first (leftmost) word on a line. You can start a command in any column, but it must always be the first word on the line.
- If you run out of space on one line for a command and its operands, you can continue it on the next line. To continue on the next line, type a comma after the last (rightmost) word on the line. You cannot split words with a continuation comma or a hyphen. When you continue a command, you can start the continued part of the command in any column, as long as it is the first (leftmost) word on the line.
- Separate operands and commands by blanks, commas, or parentheses lets them be considered separate words. Use one or more blanks to separate all words not separated by commas or parentheses. If you must use commas or parentheses, you do not have to separate words with blanks. Multiple blank spaces are treated as single blank spaces. You can type as many blanks as you want between words.
- Blank lines are ignored. You can include blank lines anywhere, in any number, even between a command and its continuation.
- Control statements are processed in the order in which they are read. Reports are printed in the order you request them.
- Any line whose first (leftmost) nonblank character is an asterisk (*) is a comment. The asterisk can be in any column. Anything after the asterisk is ignored.
- You can embed comments anywhere except between a command and its continuation lines. You cannot continue a comment with a continuation comma; to continue a comment, repeat the asterisk for each line of the comment.

Chapter 2: Planning Your Reports

This section contains the following topics:

How to Plan a Report (see page 21)
About Sample Reports (see page 22)
Types of Report Formats (see page 22)
Types of Information to Include (see page 23)
How to Further Tailor Information (see page 24)

How to Plan a Report

Before you create a report, plan what to include in your report, how you want it to look, and the time period.

To plan a report, do the following:

1. Decide whether to use short-term or long-term data for your report. For short-term data, use records extracted from the current SMF file. For long-term data, use either the collected performance data records or the archived SMF data.
2. Decide whether to use one of the Report Writer sample (predefined) reports.
3. Decide what format you want for the report: tabular, plot, graph, or historical (flashback). The format determines the primary report command that you use.
4. Decide what information you want in the report, for which resources you want this information, and how you want the information ordered and formatted. For example, you might want to create a tabular report showing all transactions, in alphabetical order, how many times each transaction was used, and the average transaction lifetime. To select this information, you use variables, variable options, ID options, and identifiers.
5. Decide what type of data you want to examine and then supply that type of data to the program.

   Note: For detailed information about the type of data each record type contains, see the *Administration Guide*.
6. Decide how to tailor your report using secondary commands. Such commands can limit the period a report covers, group the resources included in the report, or assign the report a descriptive title.

About Sample Reports

The Report Writer includes a number of predefined or sample reports. Sample reports are easy to use because the commands are already written for you. The Report Writer shows you the commands and lets you tailor them in certain ways. When you are more experienced with the Report Writer, you can write your own report commands to create reports that more precisely suit your requirements.

More information:

Sample Reports (see page 303)

Types of Report Formats

If you decide not to use a sample report, then you must write the commands to produce your report. You can generate reports in four formats:

- Tabular reports

  Tabular reports display information in a table format. Use this type of report to compare many variables at once.

- Plot reports

  Plot reports show the values of one or two variables across time. For example, you can print a report that shows the hourly CPU usage of your system.

- Graph reports

  Graph reports have a similar format to plot reports. Graph reports, however, break down resource usage by identifiers (transaction, terminal, user, and so on) within the period or periods specified. This break down can produce specific information, such as how often a certain transaction was used during peak periods.

- Historical reports

  Historical reports present a formatted listing of system events in the order they occurred. You can use historical reports to discover the source of a recent problem by using data from either the historical file or the log file.

Types of Information to Include

After you determine the format of the report, you then decide what information to include and for which resources you want the information. You specify this information as follows:

- To specify what information you want, and in what format, use variables and variable options. For example, you can create a report showing transaction use, and you can have this information presented as the number of occurrences or as percentages.
- To specify the resources about which you want information, you use ID options and identifiers.

The following sections briefly describe these topics:

- Variables
- Variable options
- ID options
- Identifiers

Variables

Use *variables* to specify what information you want to be included in a report, and in what order.

For example, the following command produces a report that plots the number of transactions executed on your system, which uses the variable TRANSACTION USE.

```
PLOT TRANSACTION USE
```

Variable Options

Use *variable options* to change either the type of data a variable represents or the format of that data. For example, by default, the variable TRANSACTION USE gives the number of transactions executed. If you want to plot the rate at which transactions were executed, you can use the variable option RATE with this variable, as follows:

```
PLOT TRANSACTION USE RATE
```

ID Options and Identifiers

Use *ID options* and *identifiers* to specify which resources you want to include in a report. You can specify ID options and identifiers with variables or with commands, as follows:

- You can limit a particular variable to certain individual resources by specifying an ID option and one or more identifiers with the variable.

  For example, the following command plots the number of times the transaction CINQ was executed, which includes the ID option TRANID and the identifier CINQ:

  ```
  PLOT TRANSACTION USE TRANID(CINQ)
  ```

- You can limit an entire tabular, historical, or graph report to resources of a given type by specifying an ID option with the report command. You can further limit the report to individual resources of that type by specifying identifiers with the ID option.

  For example, the following command generates a tabular report showing:

  - The number of times the transactions CINQ and PINQ were executed
  - The average lifetime of each

  ```
  TAB TRANID(CINQ,PINQ) TRANSACTION USE AND TRANSACTION LIFE
  ```

How to Further Tailor Information

You can further tailor the report by using secondary report commands.

For example, use the commands TO and FROM to limit a report to only those transactions that were executed between 9 and 11 a.m. on June 1. You can also use the EACH command to define the period each detail line in a report is to represent, and you can use the TITLE1, TITLE2, and TITLE3 commands to specify a multiline title for the report.

More information:

Tailor Reports Using Secondary Commands (see page 33)
Commands (see page 39)

Chapter 3: Writing a Report

This section contains the following topics:

How to Write Reports (see page 25)
Types of Reporting Commands (see page 27)
Write A Report (see page 31)
Write Multiple Reports (see page 32)
Tailor Reports Using Secondary Commands (see page 33)
Reports with a File Type of CSV (see page 34)

How to Write Reports

Before you start writing reports, follow the general process for writing and submitting any history report or series of reports.

To write a report, do the following:

1. Create a job stream. Use the following sample job stream to write reports:

```
//JOBCARD  JOB  (0000),TODD,CLASS=A,MSGCLASS=a
//REPORT   EXEC PGM=XPFRMAIN,REGION=4M
//STEPLIB  DD   DSN=sysview.REPORT.LOADLIB,DISP=SHR
//ERPTPRM  DD   DSN=sysview.REPORT.SAMPLIB,DISP=SHR
//SYSUDUMP DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//TAPSMF   DD   DISP=SHR,DSN=SMF.inputfile
//SYSIN    DD  *
INPUT(TAPSMF)
 OPTION(RECSTAT=YES)
 TAB TRAN USE
END
RUN
*
```

2. Add parameters to your EXEC statement.

   The following is a sample EXEC statement:

   ```
   // EXEC PGM=XPFRMAIN,REGION=2048K,PARM=['CASE=UPPER']
                                         ['DATAONLY=[YES]']
                                         ['DATEFRMT=[MM/DD/YY|DD/MM/YY]']
                                         ['ECHO=[YES|NO]']
                                         ['COPYRIGHT=NO']
   ```

   This EXEC statement performs the following tasks:

   - Forces all output to uppercase.
   - Prints only lines containing report data.
   - Sets the date format.
   - Specifies whether to list report commands before printing the report.
   - Prevents the copyright statement from printing.

3. Write history reporting commands.

   The following example shows the minimum commands necessary to write one report:

   ```
   primarycommand [variable]
    END
   RUN
   ```

4. Combine the history reporting commands with the job stream.
5. If you plan to use input from a tape file, mount the tape containing the file.
6. Run the job stream created in step 4.

Note: A command can begin anywhere, as long as it is the leftmost word on the line.

Types of Reporting Commands

The Report Writer uses two types of history reporting commands. You also use two special commands for the reports themselves.

- The two types of history reporting commands are as follows:
  - Global commands
  - Report commands
- Special commands for writing reports are as follows.
  - END command

    Tells the Report Writer that the report description is complete. Each report must end with the END command. After the END command, you can start another report. For each new report, the variables you set in the previous report are no longer in effect. If you want to use the same variables, you enter them again. Values set by global commands, however, remain in effect unless overridden by another command.
  - RUN command

    Starts executing the reports you described. When the Report Writer encounters RUN, it stops reading your commands and starts writing reports. Commands after the RUN command are ignored; so always make RUN the last command given to the Report Writer.

Global Commands

Global commands set values that stay in effect for all reports that are printed in one run. For example, if you use the LINECOUNT command to tell the Report Writer how many lines are on one page, then that number of lines applies to all reports in that run.

You can insert global commands anywhere among the report commands. With the exception of the GLOBAL command, global commands affect all reports in the run without regard to placement. The GLOBAL command affects only reports that come after it.

The following are the history reporting global commands:

CARDLEN

Sets the number of characters in each line the Report Writer reads. CARDLEN is useful when you want the Report Writer to ignore sequence numbers.

COUNT

Specifies the number of input records the Report Writer is to read. This command is useful for testing new reports.

DEFINE

Creates new ID options and variables.

GLOBAL

Makes a command apply to all subsequent reports. This command helps you avoid duplicating code when you create multiple reports.

GOTO

Causes the Report Writer to ignore all subsequent commands until the specified label is read.

IF

Creates new identifiers based on other identifiers, or resets existing identifiers.

INPUT

Defines the input files to use for the report.

LINECOUNT

Specifies the number of lines to print per page of the report.

OPTION

Specifies global report options, such as:

- How dates, null identifiers, and zero values are represented
- Whether summary lines, storage statistics, or dumps are produced
- Whether messages are printed

RUN

Prints the reports.

Report Commands

Report commands apply to a single report and control what happens in a single report. These commands build a report by telling the Report Writer what type of report to write, what variables to use, when to start the report, and so on.

These commands can be divided into two groups:

- *Primary* report commands, which define the type of report
- *Secondary* report commands, which alter or enhance the report

Primary Report Commands

Each report must contain one (and only one) of the following primary report commands:

FLASHBACK

Prints detailed information about each transaction executed.

GRAF

Produces a graph report that shows the value of a variable for each resource of a single type; for example, transaction, terminal, and user.

GRAF2

Produces a second graph under each graph line created by the GRAF command.

HPLOT

Produces a distribution plot of one variable against another.

INVOICE

Produces a tabular invoice report using the cost factors you specify. This report includes a separate invoice for each identifier included in the report and an optional summary for all identifiers.

MPLOT

Plots multiple variables across time.

PLOT

Plots a variable across time.

PLOT2

Creates a second plot below each plot line created by the PLOT command.

TAB

Produces a table of the values of the variables you specify.

TAB2

Produces a second table for each row of a table created by the TAB command.

VPLOT

Plots a variable vertically across time.

VPLOT2

Plots a second variable on top of a plot created by the VPLOT command.

Secondary Report Commands

Following the primary report commands, you can include secondary report commands. Secondary report commands alter or enhance the report.

For example:

- Use the secondary commands FROM and TO to limit the time range covered by a report.
- Use TITLE1 to define a title for the report.

The following are the secondary report commands:

COMMENT

Prints a comment at the bottom of each page of the report.

COST

Assigns cost values to resources for charge-back.

EACH

Defines a period from minutes to years in which to display or organize report data.

END

Indicates the end of the current report description.

EXCLUDE

Defines a time range to exclude from the report.

EXITRTN

Specifies the name of a user exit to call after each record is read and before processing.

FROM

Defines the starting date and time of the report.

GROUP

Combines resources into performance groups.

HEADER

Assigns a header to appear above the column headings in a tabular report.

INCLUDE

Defines a time range to include in the report.

PERIOD

Produces a summary of the plotted average activity of the resource.

PRTEXIT

Specifies the name of a user exit to call before each line is printed.

RANGE

Defines a value range for any variable.

SELECT

Restricts a report to the specified identifiers.

SHIFT

Limits the report to a part of each day, typically a production shift.

TITLE1, TITLE2, TITLE3

Replaces the default title lines of a report with up to three lines of text you specify.

TO

Defines the ending date and time of the report.

Write A Report

The following shows the minimum commands necessary to write one report:

```
primarycommand [variable]
 END
RUN
```

primarycommand

Defines the type of report. For example, a tabular or plot report.

variable

Specifies one or more Report Writer variables.

END

Indicates the end of the current report description.

RUN

Starts executing the report.

Write Multiple Reports

You can generate your report in more than one format. Use the following procedures to write multiple reports in a single run.

Follow these steps:

1. End each set of report commands with the END command.

   ```
   GRAF variable
   END

   PLOT variable
   END
   ```

2. Specify the RUN command following the END command for the last report.

   ```
   PLOT variable
   END
   RUN
   ```

   Your graph and plot reports are run.

Note: Each historical report must be run as a single report in a separate run. In other words, you cannot specify another primary report command (such as TAB or PLOT) in a job stream with the FLASHBACK command.

Example: Commands to Write GRAF and PLOT Reports

The following are the minimum commands to write a GRAF report and a PLOT report:

```
GRAF variable
END

PLOT variable
END
RUN
```

Each GRAF and PLOT report includes an END command.

Tailor Reports Using Secondary Commands

You can include specific information in your reports, such as shift information, by using secondary commands.

To tailor a report, insert the secondary commands before the END command of the report.

Example: Add a Shift to Your Report

The following example adds the SHIFT command to include only the hours from 8:00 a.m. to noon in the graph report.

```
GRAF variable
SHIFT 08:00:00 12:00:00
END

PLOT variable
END
RUN
```

In this example, because the SHIFT command is not used with the PLOT command, the resulting plot report covers all 24 hours. To apply the same time restrictions to the plot report as you do to the graph report, include the SHIFT command with the PLOT command.

Create Multiple Tailored Reports

You can use the same secondary command for each report you create. For example, if you are printing ten reports, you can use the SHIFT command with each primary report command to specify the period you want each report to cover. Generally, you cannot use the same secondary command twice for the same report, even if you specify two different periods. The exceptions to this rule are the COST, EXCLUDE, GROUP, INCLUDE, RANGE, and SELECT commands, which can be repeated in the same report.

Examples: Incorrect and correct use of secondary commands in a stream.

- Incorrect Command Stream

```
GRAF variable
SHIFT 08:00:00 12:00:00
SHIFT 13:00:00 17:00:00
END

PLOT variable
END

RUN
```

- Corrected Command Stream

```
GRAF variable
SHIFT 08:00:00 12:00:00
END

GRAF variable
SHIFT 13:00:00 17:00:00
END

PLOT variable
END

RUN
```

Reports with a File Type of CSV

You can save your report to a file type of CSV and download the CSV report to a PC.

After the report is on the PC, you can further process and format your reports using a PC spreadsheet program. CSV formatted reports can be only of type TAB, TAB2, and FLASHBACK.

Note: Reports with the option DATAONLY set to YES cannot be saved as a CSV formatted report.

CSVGEN Utility—Convert Generated Reports to CSV Format

You can take a TAB, TAB2, and FLASHBACK report that you have created and convert that report to CSV format for use in a PC spreadsheet program.

Note: We recommend using OPTION(ZEROFLD=ZERO) in your reports. This option helps ensure that fields with a zero value do not appear as blanks in the CSV output.

Usage of the CSVGEN utility to convert the generated report to CSV format

Use the prepared sample job $CSVJCL, which is an extension of $SAMPJCL. The CSV formatted output is generated in the second step of the $CSVJCL job:

```
//PARSE    EXEC PGM=IKJEFT01
//SYSEXEC  DD DISP=SHR,DSN=sysview.REPORT.SCSYSAMP
//SYSTSPRT DD SYSOUT=*
//RWO      DD SYSOUT=*
//CSVIN    DD DSN=*.REPORT.SYSPRINT,DISP=(OLD,DELETE)
//CSVOUT   DD SYSOUT=*
//SYSTSIN  DD *
  CSVGEN CSVIN CSVOUT VS=; DS=,
/*
```

DD names:

RWO

Copies the original (source) report here.

CSVIN

Specifies the source report, which is typically generated in the first step, but a stored report could also be used.

CSVOUT

Stores the CSV formatted report here.

SYSTSIN

Invokes the conversion utility with all parameters.

$CSVJCL introduces a second job step, converting the report generated in the first step to a CSV formatted file. CSV formatted reports contain only data and relevant headers, presented in the same order as in the source report.

CSVGEN utility has the following syntax:

```
CSVGEN {<IN> <OUT>} [OPTIONAL PARAMETERS]
```

CSVGEN

Indicates the utility name.

<IN>

(Required) Input ddname. Points to original (source) report which is being converted.

<OUT>

(Required) Output ddname. The report in CSV format is stored here.

[OPTIONAL PARAMETERS]

Some optional parameters can be used to tailor your CSV formatted output. These are only one character long.

VS=

Value Separator. Use this parameter to change the default value separator value.

Default: comma (,)

Example, to specify a semicolon use the following format:

```
VS=;
```

DS=

Decimal Separator. Use this parameter to change the default decimal separator value.

Default: period (.)

Example, to specify a comma use the following format:

```
DS=,
```

Your report is converted to CSV format and ready to download to your PC.

How to Download CSV Formatted Reports

Several ways exist to download generated CSV reports into a spreadsheet on a PC. You can store the report on your PC as a pure text file with the extension .CSV.

To download the report from the mainframe to your PC, use any of the following methods:

- (Recommended method) Use any suitable FTP client to download the report directly from the JES spool (if FTP access is enabled).
- Copy and paste the report from your terminal emulator into a text editor on your PC.
- Save the CSV report into a data set and do either of the following:
  - Use the function Download file from your terminal emulator (if it is supported).
  - Use any suitable FTP client to download the report (if FTP access to data sets is enabled).

Example: Download a CSV Formatted Report

The following example explains how to download a CSV report directly from the JES spool using Windows built-in command line FTP client. Usage is similar in different FTP clients. Bold text indicates commands that you must enter, variable parts are in italics.

```
C:\>ftp
ftp> open server.organization.com
Connected to server.organization.com.
220- IBM FTP CS V1R11 at SERVER.ORGANIZATION.COM, 11:11:11 on 2011-11-11.
220 Connection will close if idle for more than 11 minutes.
User (server.organization.com:(none)): username
331 Send password please.
Password: userpassword
230 USERNAME is logged on.  Working directory is "USERNAME.".
ftp> quote site filetype=jes
200 SITE command was accepted
ftp> quote site jesjobname=*
200 SITE command was accepted
ftp> quote site jesowner=jobownername
200 SITE command was accepted
ftp> get  JOBxxxxx.6  c:\data\csvreport.csv
200 Port request OK.
125 Sending data set USERNAME.XML#A11.JOBxxxxx.D1111111.?
250 Transfer completed successfully.
ftp: 1111 bytes received in 0,11Seconds 11,11Kbytes/sec.
ftp> bye
221 Quit command received. Goodbye.
```

Explanation of commands used in the example:

ftp

Invokes the ftp client.

open *server.organization.com*

Connects to specified server. Replace *server.organization.com* with the network address of the server you are connecting to.

username, userpassword

Specifies the user name and password. Replace *username* and *userpassword* with valid user credentials.

quote site filetype=jes

Creates the connection to JES spool files.

quote site jesjobname=*

Uses all available jobs.

quote site jobowner=*jobownername*

Selects only the job of the specified user. Replace *jobownername* with the valid user ID of the job owner.

get *JOBxxxxx.6 c:\data\csvreport.csv*

Indicates the client is ready to download the report.

JOBxxxxx.6

The client finds the report by job ID and extension. Job ID is formatted as JOB*xxxxx* where *xxxxx* is the job number. The extension (in this example, the number six after the period) tells the client which spool file to download. Spool files belonging to the specified job are numbered starting with one. So in this example the CSV report would be the sixth spool file belonging to the specified job.

c:\data\csvreport.csv

Provides the fully qualified path where you want to store the report on your computer.

bye

Closes the ftp client.

Chapter 4: Commands

This section contains the following topics:

About the Commands (see page 40)
Variables with Commands (see page 40)
ID Options with Commands and Variables (see page 40)
CARDLEN Command—Set the Character Length (see page 41)
COMMENT Command—Add a Comment Line (see page 41)
COST Command—Provides Charge-back Statistics (see page 42)
COUNT Command—Define Number of Records (see page 44)
DEFINE Command—Define ID Options and Variables (see page 45)
EACH Command—Define a Time Period (see page 54)
END Command—End a Report (see page 56)
EXCLUDE Command—Define Excluded Time Ranges (see page 57)
EXITRTN Command—Define User Exits (see page 58)
FLASHBACK AUDITLOG Command—Generate Audit Report (see page 59)
FLASHBACK LIST Command—Generate Transaction Report (see page 60)
FROM Command—Begin Time for Report (see page 62)
GLOBAL Command—Assign Parameter Values (see page 66)
GOTO Command—Jump to Specified Location (see page 67)
GRAF Command—Produce a Graph (see page 68)
GRAF2 Command—Produce Subgraphs (see page 70)
GROUP Command—Combine Resources (see page 72)
HEADER Command—Specify a Title (see page 74)
HPLOT Command—Produce a Horizontal Plot (see page 76)
IF Command—Assign Identifier Values (see page 77)
INCLUDE Command—Define Time Ranges (see page 78)
INPUT Command—Input File Name (see page 80)
INVOICE Command—Produce a Tabular Invoice Report (see page 80)
LINECOUNT Command—Set Lines Per Page (see page 82)
MPLOT Command—Produce a Vertical Plot (see page 83)
OPTION Command—Define Report Options (see page 84)
PERIOD Command—Specify a Time (see page 91)
PLOT Command—Plot Resource Usage (see page 92)
PLOT2 Command—Plot Two Variables (see page 94)
PRTEXIT Command—Define a User Exit (see page 94)
RANGE Command—Set Criteria to Include (see page 95)
RUN Command—Begin Processing and Printing Reports (see page 97)
SELECT Command—Specify Identifiers (see page 97)
SHIFT Command—Hours Covered for a Report (see page 98)
TABulate Command—Produce Tabular Reports (see page 100)
TAB2 Command—Build a Secondary Table (see page 103)
TITLE1, TITLE2, and TITLE3 Commands (see page 104)
VPLOT Command—Plot Usage in Vertical Columns (see page 106)
VPLOT2 Command—Plot Two Variables (see page 107)

About the Commands

This chapter describes Report Writer commands, including the syntax and options for each command.

For more examples of the primary commands, including sample reports, see the following chapters:

- “Tabular Reports”
- “Plot Reports”
- “Graph Reports”
- “Historical Reports”

Variables with Commands

Some commands require you to specify at least one variable.

More information:

Variables (see page 109)

ID Options with Commands and Variables

Some commands let you organize and limit the data by specifying one or more ID options. Specify ID options either with the command or with the variables specified with the command.

More information:

ID Options (see page 167)
Variables (see page 109)

CARDLEN Command—Set the Character Length

The CARDLEN command is a global command that sets the number of characters in each line that the Report Writer reads. This command is useful when you want the Report Writer to ignore sequence numbers.

Entering CARDLEN once sets the line length for all reports in the run. Place this command as the first command in your command stream; otherwise, the lines before it are treated as if they are 80 characters long.

This command has the following syntax:

```
CARDLEN nn
```

nn

- Specifies the maximum number of columns (characters) per line that the Report Writer reads.
- Must be an integer less than or equal to 80.

Default: 80

Example: CARDLEN Command

The following command specifies that the Report Writer is to read 71 characters per line:

```
CARDLEN 71
```

COMMENT Command—Add a Comment Line

The COMMENT command creates a comment line at the bottom of each page of the report.

This command has the following syntax:

```
COMMENT commentline
```

commentline

Provides the comment that you assign to the report. The comment can be up to 60 characters long. Every character in the command line must be on the same line as the COMMENT command.

COST Command—Provides Charge-back Statistics

The COST command makes charge-back statistics available for any resource.

The following list provides usage information for the COST command:

- If you use the COST command with a variable you created using the DEFINE command, then also specify the DEFERRED operand.
- Use the variable REPORT COST TOTAL to create a column showing *total* cost for each row in your report.
- Use the variable REPORT COST DISTRIBution to create a column showing *proportional total* cost for each row in your report. Use this variable only if you specified the DISTRIBution operand.
- Use the COST option with a variable to show its cost value, computed using the cost factor you specify with the COST command.

  For example, you specify a cost factor for the variable TRAN USE using the COST command. Then you could specify the variable and option TRAN USE COST with the TAB command. The created column in the report shows the cost for transactions for each identifier or period in the report.
- Use the command OPTION MONETARY=*x* to specify a one-character symbol for the monetary unit. The default setting is the dollar sign ($).

This command has the following syntax:

```
COST [variable cost-factor [idoption(identifiers)]
     [RANGE(min,max)][SELECT(name)]]
     [variable cost-factor [DEFERRED]]
     [DISTRIBution value]
```

variable

Specifies any valid variable keyword. You can use variables defined with the DEFINE command when you specify the DEFERRED operand.

cost-factor

Specifies the monetary value associated with this resource. This value can include up to five decimal positions.

idoption(identifiers)

Includes any valid ID option specified with any valid identifiers.

RANGE(*min*, *max*)

Allows multiple cost factors for the same resource based on specified value ranges.

(min, max)

Defines the value range. Specify a separate COST command for each range.

SELECT(*name*)

Applies a period specified with an INCLUDE or EXCLUDE command.

name

Specifies the name assigned by the ID operand of the INCLUDE or EXCLUDE command. This operand functions similar to the SELECT option.

For more information, see the section SELECT Option—Specifying a Time Period for a Variable (see page 202) in the chapter "Tabular Reports."

DEFERRED

By default, the cost-factor is applied to the data immediately as each record is read. Specifying DEFERRED causes the Report Writer to apply the cost factor after all the data has been read. If you specify DEFERRED, you cannot specify the *idoption*, RANGE, or SELECT operands.

DISTRIBution *value*

Creates a distributed cost report.

value

Defines the cost you want the report rows to total.

Specify the REPORT COST DISTRIBution variable to include a column in your report. This report shows the adjusted total cost for each row, with all rows adding up to the amount defined by *value*.

Example: Cost Commands

The COST commands in this example define the following cost factors:

- One cent per transaction
- 25 cents per CPU second consumed

```
COST TRAN USE    .01
COST TRAN CPU    .25

TAB TRANID TRAN USE                                  AND,
           TRAN USE  COST HDR1(TRAN USE)  HDR2(COST)  AND,
           TRAN CPU                                  AND,
           TRAN CPU  COST HDR1(TRAN CPU)  HDR2(COST)  AND,
           REPORT COST TOTAL
END
RUN
```

The TAB command uses the two cost factors, creating a tabular report with a column showing the value of each variable for each transaction. The HDR1 and HDR2 operands assign those columns appropriate headings. The last variable, REPORT COST TOTAL, creates a column that shows the sum of the costs for each row in the report.

COUNT Command—Define Number of Records

The COUNT command is a global command that defines a maximum number of input records to read for a report. After the specified number of records are read, an end-of-file condition is forced.

Entering COUNT once sets the maximum number of input records to read for each report in the run. You can use this command to experiment with new reports that can generate more output than you want.

If a FROM command is specified, the counting of records begins with the first record meeting the earliest time specified in the FROM command. Otherwise, the counting of records begins with the first record read.

This command has the following syntax:

```
COUNT nnnnnn [SELECTED]
```

nnnnnn

Provides the maximum number of records to read.

SELECTED

Counts only records that pass at least one selection criterion when determining if *nnnnnn* records have been read.

Example: Count Command

The following command specifies that the Report Writer is to stop reading records after reading 50 records that meet any selection criterion:

```
COUNT 50 SELECTED
```

DEFINE Command—Define ID Options and Variables

The DEFINE command is a global command that creates new ID options and variables for use in reports. Any variables or ID options you define are in effect for all reports in the run.

Important! When you use the DEFINE command to create variables, consider specifying the FORMAT variable option with the variable in your report command. If you do not specify the FORMAT option, the decimal point is not included in the variable values.

You can define variables and ID options from scratch, or use existing variables or ID options as a template.

This command has the following syntax:

```
DEFINE ID|XID|XVAR
```

ID

Requests the definition of a new ID option using existing ID options.

XID

Requests the definition of a new ID option.

XVAR

Requests the definition of a new variable.

More information:

Variable Options (see page 157)

DEFINE XID, XVAR—Define New ID Options or Variables

Defining new ID options or variables without using existing ID options or variables lets you report on any data field located in any supported record.

Use the following syntax to define new ID options or variables without using existing ID options or variables.

The syntax is shown with a continuation comma at the end of the first line.

```
DEFINE XID|XVAR name PRODUCT=EXPC RECORD=rectype,
                [SUBTYPE=subtype] OFFSET=data-offset
                LENGTH=data-length,[TRIPLET=triplet-offset]
                [TYPE=data-type][HDR1=header1] [HDR2=header2]
                [MIN | MAX] [FORMAT(outputformat)]
```

XID

Requests the definition of a new ID option.

XVAR

Requests the definition of a new variable.

name

Specifies the name of the new ID option or variable being created. For DEFINE XID commands, *name* must be a single word.

Note: For DEFINE XVAR commands, *name* must conform to the structure described at the beginning of the chapter "Variables (see page 109)."

PRODUCT

Identifies the product for which an ID option or variable is being defined. You can omit this operand if you use the PRODUCT operand on the OPTION command.

rectype

Specifies the record type. To specify a value in hexadecimal, prefix the value with the letter *x*, for example, RECORD=x3C. You can omit this operand if you use the RECORD operand of the OPTION command.

subtype

Specifies the record subtype.

data-offset

Specifies the offset to the data field from the beginning of the section. To specify a value in hexadecimal, prefix the value with the letter *x*, for example, OFFSET=x3C.

data-length

Specifies the length of the data field. If you specify an XVAR with a type of BINARY, 3DEC, 6DEC, or NUMERIC, the maximum length is 8 bytes.

Note: Character-type variables have no length restriction.

triplet-offset

Specifies the offset from the beginning of the section to the triplet information. To specify a value in hexadecimal, prefix the value with the letter *x*, for example, TRIPLET=x3C.

data-type

Specifies data types for the DEFINE XID or DEFINE XVAR commands.

- Specifies one of the following data types for DEFINE XID commands:

  BINARY-Indicates binary.

  CHARACTER-Indicates character.

  HEX-Indicates hexadecimal.

- Specifies one of the following data types for DEFINE XVAR commands:

  3DEC-Indicates a number with three decimal places.

  6DEC-Indicates a number with six decimal places.

  BINARY-Indicates binary. Default for DEFINE XVAR commands.

  CHARACTER-Indicates character.

  NUMERIC-Indicates decimal numbers in character format.

  HEX-Indicates hexadecimal.

  STCK-Converts a store-clock TOD field into *mm/dd/yy hh:mm:ss*.

  STCK-DATE-Converts a store-clock TOD field into *mm/dd/yy* format.

  STCK-TIME-Converts a store-clock TOD field into *hh:mm:ss* format.

  STCK-TIMEM-Converts a store-clock TOD field into *hh:mm:ss:mmm*.

header1

Specifies a string of up to eight characters to be used as the top column heading.

Default: Null string

header2

Specifies a string of up to eight characters to be used as the bottom column heading.

Default: Null string

MIN|MAX

Use the MAX and MIN options to force a new variable that by default prints a count to print the minimum or maximum value.

FORMAT

Use the FORMAT option to define the length of an output field or to specify how many decimal positions it contains.

Note: For a description of output format, see the chapter "Variable Options (see page 157)."

DEFINE ID—Define Options Using Existing ID Options

When defining a new ID option, you can include multiple character strings or existing ID options, in any order. When printed, components are concatenated without intervening spaces.

To define new ID options using existing ID options, use the following syntax:

```
DEFINE ID newid [oldid [(start,length)]] ['string']...
```

ID

Requests the definition of a new ID option.

newid

Provides the name of the new ID option you want to define.

oldid

Specifies the name of an existing ID option you want to be a component of the new ID option.

start,length

Specifies the starting position (*start*) and length (*length*) of the part of the existing ID option you want to include in the new ID option. Use these variables if you do not want to include the entire existing ID option.

string

Defines a character string that you want as a component of the new ID option. Enclose the string in single quotes. The string can be one or more characters long, and it can include or consist entirely of blanks.

DEFINE VAR—Define Variables Using Existing Variables

Use the following syntax to define new variables using existing variables:

```
DEFINE VAR newvar = expression
```

VAR

Requests the definition of a new variable.

newvar

Provides the name of the new variable you want to define.

expression

Assigns any arithmetic expression, the result of which is the value of the new variable. You can use the following to create the expression:

- Existing variables
- The MAX, MIN, TOTAL, and RANGE variable options
- Valid ID options
- Constants

Delimit all variables, operands, and parentheses with blanks. You can use the following operators in the expression:

+ = Addition

- = Subtraction

***** = Multiplication

/ = Division

! = Exponentiation

( ) = Parentheses

Examples: The DEFINE Command

The following examples show ways to enter the DEFINE command:

- Redefine the length of the ID option TRANID from eight characters to four, starting with position one:

  ```
  DEFINE ID TRANID TRANID(1,4)
  ```

- Redefine the ID option PROGID to use only four characters, starting with position 3:

  ```
  DEFINE ID PROGID PROGID(3,4)
  ```

- Create an ID option named MYKEY with components TRANID and TERMID and places a blank between the two components:

  ```
  DEFINE ID MYKEY TRANID ' ' TERMID
  ```

- Create the variable TRAN %CPUTIME by multiplying TRAN CPUTIME by 100 and then dividing by TRAN LIFETIME:

  ```
  DEFINE VAR TRAN %CPUTIME= ,
   ( TRAN CPUTIME * 100 ) / TRAN LIFETIME
  ```

  The continuation comma allows the command to be continued on a second line.

- Create the new variable DCNTMAX that yields the maximum number of transient data sets used by any one transaction:

  ```
  DEFINE VAR TRAN DCNTMAX = TRAN DCOUNT MAX
  ```

- Create the ID option KEY19, which is of record type 19, at offset X'14', of length 8, and of type hexadecimal:

  ```
  DEFINE XID KEY19 PRODUCT=EXPC RECORD=19,
             OFFSET=X14 LENGTH=8 TYPE=HEX
  ```

  The continuation comma allows the command to be continued on a second line.

- Create the new variable TERM MYVAR:

  ```
  DEFINE XVAR TERM MYVAR,
        PRODUCT=EXPC RECORD=19 OFFSET=X28,
        LENGTH=2 TYPE=HEX HDR2=myvar
  ```

  The continuation commas allow the command to be continued on subsequent lines.

Define XID Options Using the List Command

Defining a new ID Option using the LIST command lets you display any text that you specify instead of displaying the SMF record value.

Follow these steps:

1. In LIST subcommand, specify the input-type, input-length, and then continue with list of value-text pairs.

   If a value read from RECORD= matches one from the LIST=, the corresponding text is used instead of this value.

2. Use the UNKW keyword at the end of the list. UNKW specifies the text to print when the SMF record field value does not meet any of values in the list.

The Define XID option using the LIST command has the following syntax:

```
DEFINE XID name PRODUCT=EXPC RECORD=rectype,
          [SUBTYPE=subtype] [TRIPLET=triplet-offset],
          OFFSET=dataoffset [HDR1=header1],
          [HDR2=header2],
          LIST=(IT=input-type, IL=input-length,
                value1,text1,
                value2,text2,
                value3,text3,
                …
                valueN,textN,
                [UNKW,unknown])
```

For a description of the common keywords, see the section Define ID Options or Variables (see page 45).

input-type

Specifies the data type of DEFINE XID command with LIST subcommand. Specify one of the following input types:

- BIN – Indicates binary
- DEC – Indicates decimal value in PACKED format
- NUM – Indicates numeric value in character format
- CHAR - Indicates character

input-length

Specifies the length of the data field.

value1

Specifies the value in the SMF record corresponding to *text1*.

text1

Specifies text that prints in the report when value1 is met in the SMF record. Text is limited to 16 characters.

UNKW

Is a keyword, used for specify UNKW-unknown pair corresponding to all values that are not specified in the list. Notice that if UNKW-unknown pair is not specified then blank is used for all unknown values.

unknown

Specifies text that prints when the value from SMF record does not match any from the list. Text is limited to 16 characters.

Example: Using Define XID Options Using LIST Command.

This example defines the IDoption OWNER from record type 255x4.

```
DEFINE XID XOWNER PRODUCT=EXPC RECORD=4,
       OFFSET=X78 HDR1=OWNER,
       LIST=(IL=1,IT=BIN,
              0,MVS,
              1,MQSERIES,
              2,IMS,
              3,TCP/IP,
             UNKW,UNKNOWN)

TAB DATE SORT ALPHA STARTTIME SORT ALPHA XOWNER,
    COMMON STATE EXCEPTION STATUS
END
RUN
```

The following report generates:

```
           START      OWNER
  DATE     TIME                STATUS
-------- ------------ -------- --------
07/01/09 00.53.54.005 IMS      HIGH
         00.55.54.035 IMS      NORMAL
         07.25.24.438 MVS      WARNING
         07.25.24.937 MVS      WARNING
         07.25.24.951 MVS      WARNING
-------- ------------ -------- --------
-------- ------------ -------- --------
```

EACH Command—Define a Time Period

The EACH command defines a time period, from minutes to years, in which report data is organized. The effects of the EACH command depend on the report type, as follows:

- With tabular reports, use the EACH command to define the time interval you want the table to repeat. Create a DAY, DATE, or TIME column when you use the EACH command.
- For plot reports, use the EACH command to define the time period represented by each line in the plot. You can plot by the hour, day, week, or month using the EACH command. The default is one day.
- Graph reports break down activity by identifier during a given time period. If you do not specify an EACH command, the result is a single graph for the whole time period covered in the report. The EACH command produces a separate graph for each time period.

You must have the appropriate amount of data in your input file for the EACH value you specify. For example, you cannot specify EACH YEAR if your input file has data for less than one year. If the input file does not have enough data, you receive an incomplete report.

Important! The Report Writer cannot report on a smaller time interval than the data in the input file. This limitation is important when you use variables that use data collected at time intervals, such as interval variables. For example, if the input file contains data that was logged at half-hour intervals, you cannot specify an EACH value of less than 30 minutes. Generally, make the EACH value you specify the same as, or a multiple of, your system data collection interval. This warning does not apply to performance record data because such data is not collected at specified time intervals.

By default, a week is Monday through Sunday. You can set the weeks to begin on a different weekday. If you specify the EACH WEEK or EACH *nnn* WEEKs command, use the *OPTION(WEEK=weekday*) command to specify the first day of your week.

This command has the following syntax:

```
EACH [time-period | RECORD] [AVERAGED] [SUMMARIZE]
```

time-period

Defines one of the following base time units.

nnn

Specifies the number of time intervals (up to 999) in the time unit you select.

MINUTE: *nnn* minutes

HOUR: *nnn* hours

DAY: *nnn* days

WEEK: *nnn* weeks

MONTH: *nnn* months

YEAR: *nnn* years

Default: DAY

RECORD

Defines the sample interval as each record, rather than as a time interval. This option is valid only for HPLOT reports; if you specify EACH RECORD for any other report, the command is ignored.

If you specify EACH RECORD for an HPLOT report, then also use the SCALE operand to specify a scale for the HPLOT variable.

AVERAGED

Produces a report with one detail line for each period specified by the EACH command. The variable values in each line represent the average value of the variable for that period for all the input data.

For example, if you create a tabular report and specify EACH 1 HOUR AVERAGED, the report includes 24 rows. One row for each hour of the day. The first row would show the average value for each variable for the hour of midnight to 1:00 a.m. for all the days of input data.

When you use the AVERAGED operand, the EACH value must be 24 hours or less.

SUMMARIZE

Similar to the AVERAGED operand, except that the data is totaled instead of averaged. Use this operand only with variables that represent counts (not averages).

For example, if you specify EACH 1 HOUR SUMMARIZE for a tabular report, the report includes 24 rows, one for each hour of the day. The first row would show the total value for each variable for the hour of midnight to 1:00 a.m. for all the days of input data.

Examples: EACH Command

To plot the value of a variable for each half-hour period:

```
PLOT variable
  EACH 30 MINUTES
END
RUN
```

To create a distribution plot using each record as the sample interval:

```
VPLOT variable1
HPLOT variable2 SCALE(nn)
  EACH RECORD
END
RUN
```

END Command—End a Report

Use the END command to end each report. Any commands after the END command are interpreted as applying to the next report. Omitting the END command between reports causes Report Writer to issue a message that you are using multiple report commands for one report.

This command has the following syntax:

```
END
```

Example: End Command

The following is an example of the use of the END command:

```
TITLE1 REPORT1
   PLOT variable1 AND variable2
       FROM 02/01/11 00.00.01
       TO   02/15/11 23.59.59
   END

TITLE1 REPORT2
PLOT variable
   SHIFT 17:00:00 01:00:00
END
.
.
.
 RUN
```

EXCLUDE Command—Define Excluded Time Ranges

The EXCLUDE command defines time ranges to exclude from a report. You cannot use both INCLUDE and EXCLUDE commands in the same report. You can use multiple EXCLUDE commands to exclude multiple time ranges.

This command has the following syntax:

```
EXCLUDE [day] [time - time] [ID name]
        [date]
        [date time - date time]
        [month]
```

day

Specifies the day of the week. Valid values are: SUNday, MONday, TUESday, WEDnesday, THURsday, FRIday, or SATurday.

time

Specifies the time of day in *hh:mm:ss* format.

date

Specifies the date in *mm/dd/yy* format.

ID *name*

Defines a name to be associated with a specific setting of the EXCLUDE command, for use with the SELECT variable option.

For more information, see the SELECT variable option in the chapter "Tabular Reports (see page 191)" and the SELECT operand of the COST Command (see page 42) in this chapter.

month

Specifies the month. Valid values are: JANuary, FEBruary, MARch, APRil, MAY, JUNe, JULy, AUGust, SEPTember, OCTober, NOVember, or DECember

Example: EXCLUDE Command

```
EXCLUDE SUNDAY

EXCLUDE 02/04/11

EXCLUDE 12:00:00 - 13:00:00

EXCLUDE FRIDAY 16:00:00 - 17:00:00

EXCLUDE 08:00:00 - 15:59:59 ID 1STSHIFT
EXCLUDE 16:00:00 - 23:59:59 ID 2NDSHIFT

EXCLUDE JULY
```

EXITRTN Command—Define User Exits

The EXITRTN command defines a user exit to call after each record is read and before the Report Writer processes the record.

Note: One user exit, EXPGUX, is predefined for your use.

This command has the following syntax:

EXITRTN(*user-exit*)

user-exit

Defines the name of the user exit.

More information:

User Exits (see page 301)

FLASHBACK AUDITLOG Command—Generate Audit Report

The FLASHBACK AUDITLOG command:

- Generates detailed reports on event records created by the Audit Events component
- Produces one line of output per record

To change the number of columns per line, specify audit ID options with the FLASHBACK LIST command.

Logs are listed in chronological order based on the time when the audit event occurred and cannot be sorted. You can run flashback reports against the flashback file, the archive file, or a backup of the archive file (a log file).

Important! Each flashback report must be run as a single report in a separate run. In other words, you cannot specify another primary report command in a job stream with the FLASHBACK AUDITLOG command. For example, TAB, PLOT, another FLASHBACK LIST, or FLASHBACK AUDITLOG command.

This command has the following syntax:

```
FLASHBACK AUDITLOG [idoption(identifiers)]
```

***idoption*(*identifiers*)**

Includes resources specified by identifiers for an ID option. You can specify multiple ID options.

Note: The FLASHBACK AUDITLOG command can generate enormous amounts of output when run against a large input file. To avoid problems, use either the FROM and TO commands, the COUNT command, or the SELECT command to limit the output.

Example: FLASHBACK AUDITLOG Command

The following FLASHBACK AUDITLOG command is specified with the AUDTYPE ID option and identifier. This report shows only Console command records.

```
FLASHBACK AUDITLOG AUDTYPE(Console command)
END
RUN
```

A report like the following is generated:

```
                      AUDIT        SYSTEM   SUBSYS    JOB      TASK   SECURITY PRODUCT    USER   TERMINAL
  DATE     TIME       TYPE          NAME      ID      ID       NAME    GROUP   RELEASE     ID       ID
-------- -------- ---------------- -------- -------- -------- -------- -------- -------- -------- --------
05/29/09 19.59.59 Console command  CA31       GSVQ   STC05788 SCHEDULR            12.5   GSVXSCHT
05/29/09 19.59.59 Console command  CA31       GSVX   STC39650 SCHEDULR            12.0   GSVXSCHT
05/29/09 20.00.00 Console command  CA31       SYSV   STC27561 SCHEDULR            12.5   GSVXSCHT
05/29/09 20.00.00 Console command  CA31       GSVQ   STC05788 SCHEDULR            12.5   GSVXSCHT
05/29/09 20.00.00 Console command  CA31       GSVX   STC39650 SCHEDULR            12.0   GSVXSCHT
05/29/09 20.00.00 Console command  CA31       SYSV   STC27561 SCHEDULR            12.5   GSVXSCHT
-------- -------- ---------------- -------- -------- -------- -------- -------- -------- -------- --------
05/29/09 20.00.00
-------- -------- ---------------- -------- -------- -------- -------- -------- --------------  - --------
```

More information:

Historical Reports (see page 261)

FLASHBACK LIST Command—Generate Transaction Report

The FLASHBACK LIST command:

- Generates detailed reports on transactions
- Produces one line of output per transaction

You can change the number of columns per line by specifying variables with the FLASHBACK LIST command.

Transactions are listed in chronological order of ending time. You cannot sort flashback reports. You can run flashback reports against the flashback file, the archive file, or a backup of the archive file (a log file).

Important! Each flashback report must be run as a single report in a separate run. In other words, you cannot specify another primary report command in a job stream with the FLASHBACK LIST command. For example, TAB or PLOT or another FLASHBACK LIST command.

Note: Only a brief description of the FLASHBACK LIST command is given here. For a more detailed description, see the chapter “Historical Reports.”

This command has the following syntax:

```
FLASHBACK LIST [columns] [idoption(identifiers)]
 [variable [options][HDR1(text)] [HDR2(text)] [AND,
 variable [options] [HDR1(text)] [HDR2(text)]] [AND,...]]
```

columns

Defines one or more of the following, in any order, to indicate the first columns to appear in the report. Not specifying any of these columns organizes the resulting report by region ID and time.

DATETIME - Creates columns for the date and time of the record.

DATE - Creates a column for the date of the record.

TIME - Creates a column for the time of the record.

DAY - Creates a column for the day of the record (MONDAY, TUESDAY, and so forth).

JOBNAME - Creates a column for the CICS job names.

You are only able to enter one of the following: DATETIME, DATE, TIME, and DAY. The width of the output page limits the number of columns that can appear in a report. You can use one of these operands only if you specify the variables to include in your report and limit their number. The number of variables you can specify depends upon the width of the column for each.

idoption(identifiers)

Includes all the resources (or only resources specified by *identifiers*) for an ID option. You can specify multiple ID options.

variable

Indicates any Report Writer variable. If you specify a variable, also specify at least one ID option.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

HDR1*(text)*

Overrides the default top heading for a column. Replace *text* with the new column heading. To remove the header, replace *text* with a null string, for example, HDR1( ).

HDR2(*text*)

Overrides the default bottom heading for a column. Replace *text* with the new column heading. To remove the header, replace *text* with a null string, for example, HDR2( ).

AND,

Signals that another variable is specified. The comma after *AND* lets you continue to the next line.

Note: The FLASHBACK LIST command can generate enormous amounts of output when run against a large input file. To avoid problems, use either the FROM and TO commands, the COUNT command, or the SELECT command to limit the output.

Example: FLASHBACK LIST Command

The following FLASHBACK LIST command is specified with the DATETIME operand. The COUNT command, specified with the SELECTED operand, limits the report to 15 transactions meeting either the FROM or the TO criterion specified.

```
FLASHBACK LIST DATETIME TRANID TRAN LIFE AND,
                        TRAN I/OTIME    AND,
                        TRAN WAIT       AND,
                        TRAN SUSP       AND,
                        TERM I/OTIME    AND,
                        TRAN CPU
  FROM 12/06/10 21:00:00
  TO   12/06/10 22:00:00
  COUNT 15 SELECTED
END
RUN
```

FROM Command—Begin Time for Report

The FROM and TO commands specify the time period a report is to cover. The FROM command tells the Report Writer to exclude data less recent than the specified date and time. The TO command tells the Report Writer to exclude data more recent than the specified date and time. The TO date and time must be later than the FROM date and time.

How FROM and TO Relate to the Log and Historical Files

Reports cannot start earlier or end later than the data in the input file used. The dates and times that your report covers are printed at the top of each page, following the FROM and TO headings.

The FROM and TO commands relate to the log and historical files as follows:

- If the FROM and TO dates or times are outside the range of the data in the input file, the report covers the data in the file closest to the dates and times specified.
- If no FROM and TO dates are specified, the report covers all records in the input file.
- If dates but no times are specified in the FROM and TO commands, the report covers all records within the specified dates.

The FROM and TO commands have the following syntax:

```
FROM time
TO   time
```

time

Specifies the date and time, in either of two formats.

Actual Date and Time

The format for specifying an actual date and time is as follows:

```
mm/dd/yy hh:mm:ss
```

mm/dd/yy

Specifies the date in integers, *mm* is the month, *dd* is the day, and *yy* is the year. If a value is less than 10, include a leading zero; for example, specify 03/05/11 rather than 3/5/11.

Enter dates in the format *dd/mm/yy* when:

- Your operating system uses the European date format
- You specified the European date format using the DATEFRMT operand of the OPTION command

hh:mm:ss

Specifies the time in hours, minutes, and seconds. You can use periods instead of colons *(hh.mm.ss).* The hours are in military format, from 01 to 23 hours. The values for *mm* and *ss* cannot exceed 59. The default is 00:00:01.

Relative Date and Time

Specifying actual dates and times in a report you run frequently requires you to change those dates in the job every time you run it. The alternative is to use the relative date and time format. This format creates reports that start or end a certain number of minutes, hours, days, weeks, or months before the current date and time. The date and time you specify is always relative to the time that you run the report.

The relative date and time format is as follows:

```
-n [MINUTES]
 [HOURS]
 [DAYS] [AT hh:mm:ss]
 [WEEKS] [AT hh:mm:ss]
 [MONTHS]
```

-n

Specifies the number of minutes, hours, days, weeks, or months to subtract from the time at which the Report Writer starts running. The number must be an integer, prefixed by a minus sign (-). You can specify zero (-0).

MINUTES

Subtracts *n* minutes from the time at which the Report Writer starts running. The time is rounded down to the nearest minute for FROM commands, and rounded up to the nearest minute for TO commands. If *-n* is zero (-0), the Report Writer rounds the current time down to the nearest minute.

HOURS

Subtracts *n* hours from the time at which the Report Writer starts running. The time is rounded down to the nearest hour for FROM commands, and rounded up to the nearest hour for TO commands. If *-n* is zero (-0), the Report Writer rounds the current time down to the nearest hour.

DAYS

Subtracts *n* days from the current date and time. The result is rounded down to the nearest day for FROM commands, and rounded up to the nearest day for TO commands. If *-n* is zero (-0), the Report Writer rounds the current date and time down to the start of the current day.

WEEKS

Subtracts *n* weeks from the current date and time. The result is rounded down to the nearest week for FROM commands, and rounded up to the nearest week for TO commands. A week begins on Monday morning at 00:00:00. If *-n* is zero (-0), the Report Writer rounds the current date and time down to the start of the current week.

MONTHS

Subtracts *n* months from the current date and time. The result is rounded down to the nearest month for FROM commands, and rounded up to the nearest month for TO commands. If *-n* is zero (-0), the Report Writer rounds the current date and time down to the start of the current month.

AT *hh:mm:ss*

Begins (or ends) the report at the time specified by *hh:mm:ss* on the day the report is run. The value is rounded down to the nearest day or week for both FROM and TO commands. Valid only with the DAYS and WEEKS operands.

Examples: Actual and Relative Dates and Times

Actual Dates and Times

To report from December 25 2010 at 8:00 a.m. to January 1 2011, use the following commands:

```
FROM 12/25/10 08:00:00
TO   01/01/11 00:00:00
```

Relative Dates and Times

To report from the start of the current week (Monday morning at 00:00:00), use the following command:

```
FROM -0 WEEKS
```

To report from yesterday at 00:00:00 to the current time, use the following commands. The report is rounded to the last hour.

```
FROM -1 DAYS
TO   -0 HOURS
```

To report from yesterday at 8:00 a.m. to today at 8:00 a.m., use the FROM and TO commands and the AT *hh:mm:ss* operand as follows:

```
FROM -1 DAYS AT 08:00:00
TO   -0 DAYS AT 08:00:00
```

To report from yesterday at 8:00 a.m. to yesterday at 4:00 p.m., use the following commands:

```
FROM -1 DAYS AT 08:00:00
TO   -1 DAYS AT 16:00:00
```

GLOBAL Command—Assign Parameter Values

The GLOBAL command assigns values to parameters of history reporting commands. The values are applied to all remaining reports in the run until another GLOBAL command resets the value. The value set by the GLOBAL command does *not* apply to subsequent reports in which the same history reporting command appears.

This command has the following syntax:

GLOBAL *command*

command

Enter one of the following commands, specified with any of its operands. For descriptions of the operands of any of these commands, see the description of the command in this chapter.

- COMMENT
- COST
- EACH
- EXCLUDE
- EXITRTN
- FROM
- GROUP
- INCLUDE
- PERIOD
- PRTEXIT
- SHIFT
- TITLE1
- TITLE2
- TITLE3
- TO

Temporarily Override a Global Value

To override a global value for one report, issue the command before the END command for that report.

Example, to override the set GLOBAL EACH 2 HOURS within a report, specify the report command EACH 4 HOURS with the commands for that report.

Remove a Global Value

You can remove a global value by issuing a GLOBAL command without operands. For example, specifying the command GLOBAL FROM removes any global value previously set for the FROM command.

GOTO Command—Jump to Specified Location

The GOTO command causes the Report Writer to ignore all subsequent commands until the specified label is read.

This command has the following syntax:

```
GOTO .label
```

.label

A period followed by a one- to eight-character label. Specify a period and this label on a separate line in the job stream where you want the Report Writer to resume processing commands.

Example: GOTO Command

The following commands produce the sample reports CICS004 CICS.OPERSTAT and CICS016 CICS.TRANSTAT. The Report Writer ignores the commands between GOTO .MYLABEL and .MYLABEL. The sample report CICS001is not produced and no syntax error is generated for the missing operand for the EACH command.

```
CARDLEN 71
OPTIONS(ECHO=YES,RECSTAT=YES,DUMP=NO)
INPUT(TAPSMF)
OPTION(SUMMARY=YES)

* CICS004 CICS.OPERSTAT *

TAB  DATETIME OPERIDS,
      TRAN USE,
      AND TERM RESP,
      AND TRAN LIFE,
      AND TRAN MAXLIFE
TAB2  TRANID TRAN USE
END
GOTO .MYLABEL
```

```
* CICS001 CICS.CICSSTAT *

TAB JOBNAME TRAN USE                  AND,
            TRAN USE PCT              AND,
            TRAN USE RATE HDR2(RATE)  AND,
            TRAN LIFE                 AND,
            TRAN MAXLIFE              AND,
            TERM RESP                 AND,
            TRAN CPUTIME              AND,
            TRAN I/OTIME              AND,
            DLI USE                   AND,
            DLI SERVTIME
END
.MYLABEL

* CICS016 CICS.TRANSTAT *

TABULATE TRANIDS,
         TRAN USE,
         AND TRAN USE PCT HDR1(TRAN) HDR2(%USE),
         AND TRAN LIFE HDR1(TRAN) HDR2(AVE-RT),
         AND TRAN MAXLIFE HDR1(TRAN) HDR2(MAX-RT),
         AND TRAN WAITS,
         AND TERM USE     HDR1(TERM) HDR2(USE),
         AND TERM MESSAGES,
         AND TERM RESP    HDR1(TERM) HDR2(RT-AVE),

         AND DATASET USE  HDR1(FILE) HDR2(USAGE),
         AND DATASET REQTIME
END
```

GRAF Command—Produce a Graph

Only a brief description of the GRAF command is given here. For a more detailed description, see the chapter "Graph Reports (see page 247)."

The GRAF command produces a single-line bar graph of the performance of each identifier associated with a variable. For example, using the GRAF command with a TERMINAL variable produces a bar graph of the variable for each terminal.

This command has the following syntax:

```
GRAF [idoption[(identifiers)]] variable [options]
 [SCALE({n},{mX})] [SORT order] [FOR n] CHAR1(x)
```

idoption(identifiers)

Includes any ID option that is valid for the variable, specified with or without identifiers. Specifying this operand organizes the resulting report by the specified ID option and limited to the identifiers specified by *identifiers.*

If you do not specify an ID option, all identifiers of the type associated with the variable are included in the report. For example, a graph of a TRANsaction variable by default graphs the value of that variable for each transaction.

Specifying the ID option and identifier TRANID(TRAN1) graphs the value of the variable for transaction TRAN1 only. Specifying the ID option TERMID graphs the value for each terminal.

variable

Indicates any Report Writer variable.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

SCALE(*n*,*m*X) or SCALE(*n*) or SCALE(*m*X)

By default, the highest value of the axis of a bar graph is set to the highest value of the graphed variables. For variables that represent percentages, the default maximum value is 100 percent.

To override this dynamic scale, specify *n* as the maximum value. If the variable that you use produces integers, *n* must be an integer. You cannot use more than three digits to the right of the decimal point when specifying *n.*

In some cases, bar graphs in a graph Report are too short to appear. To display these graphs, you can cause each plot character to appear as *m* characters by specifying the multiplier *m*X, where *m* is an integer. Because the resulting bar graphs no longer correspond numerically to the scale, *m*X is useful only for determining relative performance.

SORT *order*

Sorts the report. Replace *order* with one of the following:

- (Default) DESCENDING

  Sorts identifiers in descending order of value of the variable specified.

- ASCENDING

  Sorts identifiers in ascending order of value of the variable specified.

FOR *n*

Limits the report to the *n* identifiers in each time period that would appear first in the report, *n* must be an integer. For reports sorted in descending order, these limits are the *n* most active identifiers. For reports sorted in ascending order, these limits are the *n* least active identifiers.

CHAR1(*x*)

Specifies the plot character. Replace *x* with any valid EBCDIC character.

GRAF2 Command—Produce Subgraphs

Only a brief description of the GRAF2 command is given here. A more detailed description can be found in the chapter “Graph Reports.”

Use the GRAF2 command to produce one or more subgraphs for each individual resource included by the GRAF command. These subgraphs can do either of the following:

- Graph the performance of a second, related set of resources.

  For example, if your GRAF command graphs by terminal, a GRAF2 command could do the following:

  - List each transaction used by each terminal
  - Show the average terminal response time for each

- Graph the values of one or more variables for each resource.

  For example, if your GRAF command graphs by transaction, a GRAF2 command could show the usage, CPU use, and average lifetime of each transaction.

You cannot use the GRAF2 command without the GRAF command.

This command has the following syntax:

```
GRAF2 [VARS|idoption[(identifiers)]] variable [options]
 [HDR(text)][CHAR1(x)] [SCALE({n},{mX},{COMMON})] [SORT order]
 [FOR n] [AND,|[variable [options] [HDR(text)] [CHAR1(x)]] [AND,...]
```

VARS

Indicates that there is more than one variable in this GRAF2 report.

***idoption*(*identifiers*)**

Includes any ID option that is valid for the variable, specified with or without identifiers. Specifying this operand organizes the resulting subgraphs by the specified ID option and limited to the identifiers specified by *identifiers.* If you do not specify an ID option, all identifiers of the type associated with the variable are included in the subgraphs. You can use this operand only if you specify a single variable with the GRAF2 command.

variable

Indicates any Report Writer variable.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

HDR*(text)*

Overrides the default heading for the variable. Replace *text* with a heading of 15 or fewer characters.

CHAR1(*x*)

Specifies the plot character for the variable. Replace *x* with any valid EBCDIC character.

SCALE(*n*,*m*X) or SCALE(*n*) or SCALE(*m*X) or SCALE(COMMON)

By default, the highest value of a bar graph axis is set to the highest value of the graphed variables. For variables that represent percentages, the default maximum value is 100 percent.

To override this dynamic scale, specify *n* as the maximum value. If the variable that you use produces integers, *n* must be an integer. You cannot use more than three digits to the right of the decimal point when specifying *n.*

In some cases, bar graphs in a graph report are too short to appear. To display these graphs, you can cause each plot character to appear as *m* characters by specifying the multiplier *m*X, where *m* is an integer. Because the resulting bar graphs no longer correspond numerically to the scale, this graph is useful only for determining relative performance.

To have the same scale used for both variables, specify SCALE(COMMON). You can specify the SCALE operand only with the first variable you specify with the GRAF2 command.

SORT *order*

Sorts the report. Replace *order* with one of the following:

- (Default) DESCENDING

  Sorts identifiers in descending order of value of the variable specified.

- ASCENDING

  Sorts identifiers in ascending order of value of the variable specified.

You can specify the SORT operand only with the first variable you specify with the GRAF2 command.

FOR *n*

Limits the report to the *n* identifiers in each time period that would appear first in the report, where *n* is an integer. For reports sorted in descending order, these limits are the *n* most active identifiers. For reports sorted in ascending order, these limits are the *n* least active identifiers. You can specify the FOR operand only with the first variable you specify with the GRAF2 command.

AND,

Signals that another variable is being specified. The comma after *AND* lets you continue to the next line. If you specify multiple variables, use the VARS operand.

Example: GRAF2 Command

The following commands produce a report showing the following:

- The maximum lifetime for the ten transactions with the longest maximum lifetimes each day
- The 20 highest maximum terminal response times for each

```
GRAF  TRANSACTION MAXLIFE FOR 10 SCALE(10.0)
 GRAF2 TERMINAL MAXRESP    FOR 20 SCALE(10.0)
 EACH DAY
END
RUN
```

GROUP Command—Combine Resources

The GROUP command lets you combine resources into performance groups. You can use GROUP with the TAB command, the GRAF command, or any of the plot commands.

This command has the following syntax:

```
GROUP idoption group-name(identifiers) [INCLUDE] [RANGE(rangeid)]
```

idoption

Defines an ID option identifying the type of resources to include in the group being defined. For example, specify the ID option TRANID if the group is to consist of transactions defined by transaction IDs.

group-name(*identifiers*)

Defines any 1- to 44-character name. If you use blanks in the group name, enclose the group name in single quotes. Replace *identifiers* with the identifiers to include in the group (for example, transaction IDs), separated by commas.

You can use generic characters in the identifiers. Once you define a group, you can use the group name as you would any predefined Report Writer ID option. That is, you can use a group name together with commands or variables to do the following:

- Organize the data in a tabular or graph report by group.

  For example, You are asked to produce a report with a set of specific transactions. Define the desired transactions with the group name MYTRANS and then specify MYTRANS as an ID option with the TAB or GRAF commands.

- Limit the value of a variable to the activity of one or more members in the group. For example, limit the variable value to the transactions in the group PAYROLLTRANS beginning with *AC* or to all transactions in the group.

INCLUDE

Includes this group in the report even if it is not included with the identifiers specified with the TAB command. The value for this group is not included in the average or total value.

RANGE(*rangeid*)

Tells the Report Writer to include in the group only those resources that fall within the range of the range ID. Replace *rangeid* with the ID assigned with the RANGE command that assigns the range criteria that you want members of the defined group to meet.

Examples: Group Command

- The following commands define groups by terminal ID and produce a report showing the average response time of each group:

```
GROUP TERMID DIVISION.WESTERN(L1*)
GROUP TERMID DIVISION.EASTERN(L2*)
GROUP TERMID DIVISION.NORTHERN(L3*)
GROUP TERMID DIVISION.SOUTHERN(L4*)

TAB TERMID(DIVISION*) TERM RESP
END
RUN
```

- The following commands define groups by ranges of transaction lifetime and produce a report showing the average use and lifetime of each group:

```
GROUP TRANID RANGE:00.000-10.000(*)  RANGE(TLIFE1)
 GROUP TRANID RANGE:10.001+(*)        RANGE(TLIFE2)

RANGE TRAN LIFE 0-10     ID TLIFE1
RANGE TRAN LIFE 10-99999 ID TLIFE2

TAB TRANID(RANGE*) TRAN USE AND TRAN LIFE
END
RUN
```

HEADER Command—Specify a Title

The HEADER command specifies a title to appear in a tabular report above the column headings and to the right of the PERIOD field.

The HEADER command is valid only with tabular reports. You can specify multiple HEADER commands for one tabular report.

This command has the following syntax:

```
HEADER(nn,mm) text
```

nn

Defines the print column where you want the first character of your heading to appear. Because headings appear on the same line as the PERIOD field, *nn* must be 20 or greater.

mm

Defines the length of the heading.

text

Provides the heading you want to appear in your report. Do not use string delimiters.

Note: You can use the RIGHT variable option to shift variable columns to the right so they do not begin before print column 20. Column 20 is where headings controlled by the HEADER command can begin. For more information, see the chapter "Tabular Reports (see page 191)."

Example: HEADER Command

The following commands create a report showing total transaction statistics, plus statistics on two particular transactions:

```
HEADER(28,13) -----EXPC----
HEADER(46,13) -----CSSN----

TAB TIME TRAN USE              AND,
 TRAN LIFE             AND,
 TRAN USE  TRANID(EXPC) AND,
 TRAN LIFE TRANID(EXPC) AND,
 TRAN USE  TRANID(CSSN) AND,
 TRAN LIFE TRANID(CSSN)
 FROM 02/03/08 08:00:00
TO   02/03/08 13:00:00
EACH HOUR
END
RUN
```

The previous commands create a report like the following:

```
SHIFT  NONE
EACH   HOUR
PERIOD NONE                -----EXPC----      -----CSSN----
         TRAN     TRAN     TRAN     TRAN      TRAN     TRAN
TIME     USE      LIFE      USE     LIFE       USE     LIFE
----- -------- -------- -------- -------- -------- --------
08.00      97    144.3       37    0.233       11    0.153
09.00      69    176.4       22    0.343        5    0.012
10.00     143    51.17      109    0.070        4    0.494
11.00      93    90.58       29    0.531       21    0.071
12.00      41    178.1        9    0.688        1    0.025
----- -------- -------- -------- -------- -------- --------
PERIOD    443    111.1      206    0.220       42    0.125
----- -------- -------- -------- -------- -------- --------
```

HPLOT Command—Produce a Horizontal Plot

The HPLOT command plots the distribution of one variable based on the value of another variable. The variable specified with the HPLOT command plots along the horizontal axis. The second variable, specified with either the VPLOT or the MPLOT command, plots along the vertical axis.

Use the HPLOT command with either the MPLOT or VPLOT command. The HPLOT command must follow the MPLOT or VPLOT command.

To set the plot character for an HPLOT report, specify the CHAR1 operand with the MPLOT or VPLOT command.

This command has the following syntax:

HPLOT *variable* [*options*] [SCALE(*n*)] [SCATTER]

variable

Indicates any Report Writer variable.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

SCALE(*n*)

Overrides the default maximum plot scale.

By default, the highest value of the horizontal axis of a plot is set to the highest value of the graphed variables. For variables that represent percentages, the default maximum value is 100 percent.

To override this dynamic scale, specify *n* as the maximum value. If the variable that you use produces integers, *n* must be an integer. You cannot use more than three digits to the right of the decimal point when specifying *n.* Use SCALE(*n*) with the RECORD option of the EACH command. Specifying a value for *n* appropriate for the activity you are plotting.

SCATTER

Prints a scatter plot showing how many times each point in the plot is reached.

Example: HPLOT Command

The following commands plot transaction use along the vertical axis against terminal response time along the horizontal axis, for each 15-minute interval:

```
VPLOT TRAN USE CHAR1(+)
HPLOT TERM RESP SCALE(3.000)
EACH 15 MINUTES
END
RUN
```

IF Command—Assign Identifier Values

The IF command lets you assign new values to identifiers or create new identifiers based on existing identifiers.

This command has the following syntax:

```
IF idoption1=(identifier[,identifier,identifier...])
THEN idoption2=newvalue [ELSE idoption3=newvalue]
```

idoption1

Specifies the ID option whose value is being tested.

identifier

Specifies an ID option or one or more identifiers to compare to *idoption1*.

idoption2

Specifies the ID option whose identifiers are assigned a new value when the IF clause is true. This value can be the same as *idoption1.*

newvalue

Defines the new value to assign to the identifier. Replace *newvalue* with either an ID option or a character string. If you specify a character string, enclose it in single quotes.

idoption3

Specifies the ID option whose identifiers are assigned a new value when the IF clause is false. This value can be the same as *idoption2.*

Examples: IF Command

The following command replaces the value of TERMID with the value of TRANID whenever the value of TERMID is equal to *L**:

```
IF TERMID=(L*) THEN TERMID=TRANID
```

The following command replaces the value of TRANID with the character string *production* whenever the value of TRANID begins with *P*:

```
IF TRANID=(P*) THEN TRANID='production' ELSE TRANID='test'
```

The following command creates the ID option DEPARTMENT. Whenever the value of TRANID begins with *P*, DEPARTMENT is assigned the value *production*.

```
IF TRANID=(P*) THEN DEPARTMENT='production'
```

INCLUDE Command—Define Time Ranges

The INCLUDE command defines time ranges to include in a report. You cannot use both INCLUDE and EXCLUDE commands in the same report. You can use multiple INCLUDE commands to include multiple time ranges.

This command has the following syntax:

```
INCLUDE [day] [time - time] [ID name]
 [date]
 [date time - date time]
 [month]
```

day

Specifies the day of the week specified as: SUNday, MONday, TUESday, WEDnesday, THURsday, FRIday, or SATurday

time

Specifies the time of day in *hh:mm:ss* format

date

Specifies the date in *mm/dd/yy* format

ID *name*

Specifies an identifying name to be associated with a specific setting of the INCLUDE command, for use with the SELECT variable option.

month

Specifies the month specified as: JANuary, FEBruary, MARch, APRil, MAY, JUNe, JULy, AUGust, SEPTember, OCTober, NOVember, or DECember

Examples: INCLUDE Command

```
INCLUDE SATURDAY

INCLUDE SUNDAY

INCLUDE 07/04/08

INCLUDE 12:00:00 - 13:00:00

INCLUDE FRIDAY 16:00:00 - 17:00:00

INCLUDE 08:00:00 - 15:59:59 ID 1STSHIFT
INCLUDE 16:00:00 - 23:59:59 ID 2NDSHIFT
INCLUDE 00:00:00 - 07:59:59 ID 3RDSHIFT

INCLUDE JULY
```

INPUT Command—Input File Name

The INPUT command specifies what file name is used for input to the Report Writer.

This command has the following syntax:

`INPUT(`*`filename`*`,UNBLOCKED)`

filename

Assigns the filenames of the input files as specified in the DD statements.

UNBLOCKED

Includes unblocked records.

Default: *TAPSMF*

Note: The Report Writer expects input from tape and processes the file as if you had specified the file name TAPSMF if you either:

- Do not specify a file name using the INPUT command, or
- Specify a file name using the INPUT command that does not match any name defined to the Report Write

INVOICE Command—Produce a Tabular Invoice Report

The INVOICE command produces a tabular invoice report using the cost factors you specify. A separate invoice is created for each identifier included in the report.

Each invoice:

- Begins on a new page
- Includes a row for each variable you specify
- Shows the value of the variable for the identifier and the cost associated with it

When using the INVOICE command, use the COST command to specify a cost factor for each variable you want included in the invoice. Then, following the INVOICE command, specify each variable twice, once without the COST option and once with it.

You can include a summary page showing the total value and cost for each variable for all identifiers in the report. To include this information, specify the variable REPORT COST TOTAL as the last variable with the INVOICE command.

This command has the following syntax:

```
INVOICE [idoption[(identifiers)]] variable [options] [HDR(text)] [SORT order]
 [AND,variable [options] [HDR(text)]] [AND,...]
```

***idoption*(*identifiers*)**

Includes any ID option that is valid for the variables you specify. You can specify one or more identifiers (separated by commas) to limit the report to a subset of the identifiers associated with the ID option. You can use generic characters when specifying *identifiers*.

variable

Includes any Report Writer variable. Each invoice in the report includes a row for each variable you specify.

options

Specifies any valid variable options. To get complete invoices, specify each variable once without the COST option and once with it. For more information about the COST option, see the COST Command (see page 42) in this chapter.

HDR(*text*)

Overrides the default description for the row for a variable. Replace *text* with the new description.

SORT *order*

Sorts the report. Replace *order* with one of the following:

- (Default) DESCENDING

  Sorts variables in descending order. The order starts with the variable specified with the SORT operand, and, when the value is the same for two entries, from left to right.

- ASCENDING

  Sorts variables in ascending order. The order starts with the variable specified with the SORT operand, and, when the value is the same for two entries, from left to right.

- ALPHA

  Sorts variables in alphanumeric order with respect to the specified identifier; for example, by transaction or terminal.

AND,

Signals that another variable is being specified. The comma after *AND* lets you continue to the next line.

Example: INVOICE Command

The following commands create an INVOICE report containing one page for each of the transaction identifiers. Because the variable REPORT COST TOTAL is specified, the report also includes a summary page showing the total costs for all transactions.

```
INPUT(TAPEIN)
  COUNT 400
  COST TRAN USE  .01
  COST TRAN TCPU .25
  INVOICE TRANID TRAN USE  AND,
                 TRAN LIFE AND,
                 TRAN USE  COST AND,
                 TRAN TCPU COST AND,
                 REPORT COST TOTAL
  END
  RUN
```

LINECOUNT Command—Set Lines Per Page

The LINECOUNT command sets the number of printed lines allowed per report page.

The LINECOUNT command is a global command. Entering it once sets the line count for all reports. To avoid printing across page creases, make LINECOUNT one of the first commands in your list of commands.

This command has the following syntax:

```
LINECOUNT n
```

n

Defines the maximum number of printed lines per page. The minimum value is 20, and the maximum value depends on the size of the paper, but cannot exceed n = 999.

Default: 60

Example: LINECOUNT Command

Set the line count at 72 lines per page, use the following command:

```
LINECOUNT 72
```

MPLOT Command—Produce a Vertical Plot

The MPLOT command produces a vertical plot similar to the plot produced by the VPLOT command. However, MPLOT only prints the outline of the detail lines. You can specify as many variables per MPLOT report as you want.

This command has the following syntax:

```
MPLOT variable [options] [SCALE({n},{mX})] [CHAR1(x)] [HDR1(text)][HDR2(text)]
 [HDR(text)] [AND,
 variable [options] [SCALE({n},{mX})][CHAR1(x)] [HDR1(text)] [HDR2(text)]
 [HDR(text)]] [AND,...]
```

variable

Indicates any Report Writer variable.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

SCALE(*n*) or SCALE(*m*X) or SCALE(*n*,*m*X)

Overrides the default maximum plot scale.

By default, the highest value of the plot horizontal axis is set to the highest value of the graphed variables. For variables that represent percentages, the default maximum value is 100 percent.

To override this dynamic scale, specify *n* as the maximum value. If the variable that you use produces integers, *n* must be an integer. You cannot use more than three digits to the right of the decimal point when specifying *n.*

You can also widen the outline of the plot by using the multiplier *m*X, where *m* is an integer.

CHAR1(*x*)

Specifies the plot character. Replace *x* with any valid EBCDIC character.

HDR1(*text*)

Overrides the default top heading for the scale of the plot. Replace *text* with the new heading. To remove the header, replace *text* with a null string, as in HDR1( ).

HDR2(*text*)

Overrides the default bottom heading for the scale of the plot. Replace *text* with the new column heading. To remove the header, replace *text* with a null string, as in HDR2( ).

HDR(*text*)

Overrides the default description next to the SYMBOLS heading that describes the plotted variables. Replace *text* with the new description, not to exceed 30 characters.

AND,

Signals that another variable is being specified. The variable is plotted on top of the preceding variables. The comma after AND lets you continue to the next line.

OPTION Command—Define Report Options

The OPTION command is a global command that defines cosmetic options for all reports. Entering it once sets options for all reports.

This command has the following syntax:

```
OPTION ([AVERAGE=[NO | YES | n ]]
                [DEFERRED | DNW | IMMEDIATE]],
        [CASE=UPPER],
        [DATAONLY=[NO | YES]],
        [DATEFRMT=[MM/DD/YY | DD/MM/YY]],
        [DEFEREXT=[NO | YES]],
        [DUMP=[NO | YES]][CORRUPT],
        [ECHO=[YES | NO]],
        [HEXID=[NO | YES]],
        [IOERR=[NO | YES | BYPASS]],
        [JOBZONE=[nWEST | nEAST]],
        [MAXTO=[YES | NO]],
        [MONETARY=n],
        [NULLID=[BLANK | 'xxxx' | n/a]],
        [NULLINE=[YES | NO]],
        [PERIOD=WEIGHTED]],
        [PERIOD=[NO | YES]],
        [PGMSTOR=[NO | YES]],
        [PRODUCT=EVSE|EXPC|ESQL|EXPV],
        [PRINT=GROUP],
        [RANGE=WEIGHTED],
        [RECAP=[NO | YES],
        [RECORD=n],
        [RECSTAT=[NO | YES]],
        [RPTSTAT=[NO | YES | xxxxx]],
        [STORAGE=[BELOW | ABOVE]],
        [STORSTAT=[NO | YES],
        [SUMMARY=YES | NO | n],
```

```
[TAB2SKIP=[n | PAGE],
[TRACE=[NO | STORAGE | LOADS],
[WEEK=weekday],
[XMSG=[nnn | ALL],
[ZEROFLD=[BLANK | ZERO | 'xxxx']]
['SRT=nnn']
['COPYRIGHT=NO'])
```

AVERAGE

Specifies whether to print a line giving averages of the items listed in the report. If you set AVERAGE=*n,* the averages are printed only if the report contains *n* or more detail lines. For graph reports, you can specify when the average is calculated and if it is weighted.

AVERAGE=IMMEDIATE calculates the average line during data accumulation. The average is weighted.

AVERAGE=DEFERRED calculates the average line before printing the report. The average is weighted. Use this setting with the FOR operand or RANGE DEFERRED command and want the average line to reflect only the lines being printed.

AVERAGE=DNW (Deferred Not Weighted) calculates the average line before printing the report, but the average is not weighted.

AVERAGE=NO (no average line).

Default: IMMEDIATE (for graph reports)

CASE

Changes the case of the report.

Specify CASE=UPPER to force all output to uppercase. You can also use the PARM= parameter of your // EXEC statement to perform this function.

Default: Output is in mixed case.

Note: For more information, see the chapter "Writing a Report (see page 25)."

DATAONLY

Prints only lines containing report data when you specify YES. Otherwise, all lines are printed. You can also use the PARM= parameter of your // EXEC statement to perform this function.

Default: NO

Note: For more information, see the chapter "Writing a Report (see page 25)."

DATEFRMT

Sets the format of dates in report output to *mm/dd/yy* or *dd/mm/yy*. You can also use the PARM= parameter of your // EXEC statement to perform this function.

Default: MM/DD/YY

Note: For more information, see the chapter "Writing a Report (see page 25)."

DEFEREXT

Specifies when to call a user exit.

DEFEREXT=NO passes control to the exit before any processing is done with the current record.

DEFEREXT=YES defers the user exit call. The call processes when all selection criteria have been applied and passes to the user exit only those records passing all selection criteria. For example, ID options and RANGE commands.

For example, the ID options and RANGE commands.

Default: NO

DUMP

Specifies whether to print a diagnostic dump of input records.

DUMP=YES dumps all used records.

DUMP=CORRUPT dumps all records with detected corruption.

Default: NO

ECHO

Specifies whether to list your report commands before printing the report. Set the value to NO only after you verify the report commands. Any report commands preceding the ECHO=NO command in the job stream appears with the report. You can also use the PARM= parameter of your // EXEC statement to perform this function.

Default: YES

Note: For more information, see the chapter "Writing a Report (see page 25)."

HEXID

Specifies whether to allow nonprintable characters in identification fields.

HEXID=NO treats any identification field that contains a nonprintable character as invalid. All information regarding the resource is saved and printed under the identifier INVALID.

HEXID=YES deactivates the test for nonprintable characters. When the report prints, however, all nonprintable characters are printed as periods.

Default: YES

IOERR

Determines what happens if the Report Writer encounters an I/O error in input data.

IOERR=NO causes the Report Writer to stop and no reports are printed.

IOERR=YES causes the Report Writer to treat the error as an end-of-file and produces reports defined up to that point.

IOERR=BYPASS causes the Report Writer to attempt to bypass the error condition and continue processing input.

Default: NO

JOBZONE

Sets the job zone (the number of hours east or west of Greenwich Mean Time). Specify *n*EAST or *n*WEST, where *n* is an integer.

Default: Value set by the ZONE parameter of the IPL SET command.

MAXTO

Applies only if every report in the run includes a TO command. By default, an end-of-file condition is forced when the maximum TO time specified with any report is exceeded. If the input file is not in chronological order, specify MAXTO=NO to force the entire file to be read.

Default: YES

MONETARY

Specifies the character to precede currency numbers. You can specify any single character.

Default: $ (dollar sign)

NULLID

Defines a four-character string to print in any identifier field for which no identifier exists. Specify NULLID=BLANK to print blanks in fields for which no identifier exists.

Default: The characters n/a

NULLINE

Specifies whether to print lines with all zero values.

Default: YES for plot reports; NO for tabular reports

PERIOD

Controls whether the period line of a plot report is a weighted value; that is, the average of all the detail lines. PERIOD=WEIGHTED converts the period line into a weighted value that represents a true average over the period.

Default: Not weighted

PERIOD

Controls whether to print the period line of a tabular report.

Default: YES

PGMSTOR

Controls whether the diagnostic storage analysis facility displays the storage acquired by each Report Writer internal program.

Default: NO

PRODUCT

Specifies the product for DEFINE XID or DEFINE XVAR commands. Specify XPFC for Command for CICS.

Default: No default

PRINT

Controls whether a report prints only group items.

Default: GROUP

RANGE

Controls whether the range line of a plot report is a weighted value; that is, the average of all the detail lines. RANGE=WEIGHTED converts the range line into a weighted value.

Default: Not weighted

RECAP

Specifies whether to produce a summary value for each identifier in the report. Depending on the variable, the summary line includes the total, average, maximum, or minimum values of the variable. This option is valid only for tabular reports organized by identifier and time.

RECORD

Specifies the decimal value of the record type for DEFINE XID or DEFINE XVAR commands.

Default: No default

RECSTAT

Specifies whether to print statistics about CA SYSVIEW records following your report. RECSTAT=YES prints information including the length of each record and each of its components (in the case of performance records), the percentage of the total disk space that each record or component comprises, and the time period covered by each.

This helps you determine which CICS resources to collect data for. In particular, if a given component contains a great deal of data, but you seldom run reports on this data, you may choose to stop collecting that data in order to lessen the storage required for the SMF files.

Default: NO

RPTSTAT

Controls how often the following message is issued to the console:

ERPT274I STATUS: CNT - *nnnnn.*

TIME - *mm/dd/yy hh:mm:ss*

nnnnn provides the current number of input records processed. *mm/dd/yy hh:mm:ss* provides the timestamp of the current record.

RPTSTAT=YES produces a status message every 10,000 records. RPTSTAT=*nnnnn* produces a status message every *nnnnn* records. You can also specify HOUR, DAY, or WEEK to generate the status message on a hourly, daily, or weekly interval.

Default: NO

STORAGE

If 31 bit addressing is available, specifies whether storage is acquired above or below the 16 MB line.

Default: ABOVE

STORSTAT

Specifies whether to print storage statistics following the last report. STORSTAT=YES prints the storage amounts allocated to each report and the total amount of unused virtual storage.

Default: NO

SUMMARY

Pertains only to tabular reports using an ID option. SUMMARY=YES prints a summary line. If you specify SUMMARY=*n*, summary lines are printed only if the report contains at least *n* lines. Depending on the variable, the summary line includes the total, average, maximum, or minimum values of the variable.

Default: YES

TAB2SKIP

For TAB2 reports only, controls how much space is left between identifiers listed by the TAB command. Either specify TAB2SKIP=*n*, where *n* is the number of lines to be skipped, or specify TAB2SKIP=PAGE to generate a page break after each identifier.

Default: 0 (no space)

TRACE

Controls the diagnostic tracing facility. TRACE=STORAGE causes message ERPT267I to print every time an internal storage request is processed. This message shows the name of the internal program requesting storage and shows how much storage the program is requesting.

TRACE=LOADS causes message ERPT283I to print whenever a Report Writer module is loaded or freed. This message shows the name of the module.

Default: NO

WEEK

Specifies the first day of the week (SUNday, MONday, TUESday, WEDnesday, THURsday, FRIday, or SATurday). If you specify the EACH *nnn* WEEKs command, you can use this operand to specify whether a week is Monday through Sunday, Tuesday through Monday, and so on.

Default: MONday

XMSG

Excludes messages from printing and from displaying on the console. XMSG=*nnn* excludes message ERPT*nnn* from output. XMSG=ALL excludes all messages.

Default: No messages are excluded.

ZEROFLD

Defines a value to replace any zero values. ZEROFLD=BLANK replaces any zero value with a blank. ZEROFLD=ZERO prints a zero. ZEROFLD=*xxxx* replaces each zero value with *xxxx.*

Default: BLANK for tabular reports; ZERO for graph and plot reports.

SRT

Specifies a new default decimal value for the CA SYSVIEW record type.

For example, specifying SRT=250 changes the default record type from 255 to 250.

Default: 255

Note: The CA EXPLORE Report Writer automatically recognizes whether CA SYSVIEW produced the SMF record. Using the SRT option disables automatic recognition.

COPYRIGHT

Controls whether to print the copyright statement lines. Specifying COPYRIGHT=NO prevents the copyright statement from printing. Notice that first statement is printed before the options are processed. To stop the first statement from printing, include the EXEC parameter of COPYRIGHT=NO.

Note: For more information, see the chapter "Writing a Report (see page 25)."

Example: OPTION Command

The sample OPTION command does the following:

- Prints a zero for each zero value in the report
- Prints lines with all zero values
- Prints a line showing the average values of the items in the report when the report contains five or more detail lines

```
OPTION(ZEROFLD=ZERO,NULLINE=YES,AVERAGE=5)
```

PERIOD Command—Specify a Time

The PERIOD command produces one report per time-period, including a summary of the average activity of variables during a specified time. The PERIOD command can be used with the PLOT and TAB commands.

You must have the appropriate amount of data in your input file for the period you specify. For example, you cannot specify PERIOD YEAR if your input file has less than one year of data.

This command has the following syntax:

```
PERIOD time-period [SKIP=n]
```

time-period

Specifies one of the following base time units.

nnn

Specifies the number of time intervals (up to 999) in the time unit you select.

Default: 1

MINUTE = *nnn* MINUTES

HOUR = *nnn* HOURS

DAY = *nnn* DAYS

WEEK = *nnn* WEEKS

MONTH = *nnn* MONTHS

YEAR = *nnn* YEARS

SKIP=*n*

Defines the number of lines to skip between reports for each period. *n* can range from 1 to 9. If you do not specify SKIP=*n,* a page break occurs between periods.

Examples: PERIOD Command

To request a summary of the average daily activity of a resource, use the following commands:

```
PLOT variable
PERIOD DAY
END
RUN
```

For a summary of the average activity of a resource every three months, use the following commands:

```
TAB variable
PERIOD 3 MONTHS
END
RUN
```

PLOT Command—Plot Resource Usage

The PLOT command produces reports that plot resource usage horizontally across time.

This command has the following syntax:

```
PLOT variable [options] [ALONE] [SCALE({n},{mX})] [CHAR1(x)]) [CHAR2(x)]
```

variable

Indicates any Report Writer variable.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

ALONE

Prevents the display of the global (systemwide) value of the variable. To use this operand, restrict the variable with an ID option.

SCALE(*n*) or SCALE(*m*X) or SCALE(*n*,*m*X)

Overrides the default maximum plot scale.

By default, the highest horizontal axis value of a plot equals the highest value reached by a variable in the plot. Or, for variables representing percentages, 100 percent. To override this dynamic scale, specify *n* as the maximum value.

If the variable that you use produces integers, *n* must be an integer. In some cases, bar graphs in a plot report are too short to appear. To display these graphs, you can cause each plot character to appear as *m* characters by specifying the multiplier *m*X, where *m* is an integer. Because the resulting bar graphs no longer correspond numerically to the scale, this graph is useful only for determining relative performance.

CHAR1(*x*)

Specifies the plot character that represents the restricted variable. Replace *x* with any valid EBCDIC character.

CHAR2(*x*)

Specifies the plot character that represents the global variable. Replace *x* with any valid EBCDIC character.

Example: PLOT Command

The following commands plot the value of the variable TRAN LIFE for transaction CINQ and the value for all transactions. The plot uses a scale of 0 to 500. The character *X* is used to plot the activity of transaction CINQ and a plus sign (+) is used to plot the activity of all transactions.

```
PLOT TRAN LIFE TRANID(CINQ) SCALE(500) CHAR1(X) CHAR2(+)
END
RUN
```

More information:

Plot Reports (see page 225)

PLOT2 Command—Plot Two Variables

The PLOT2 command is used with the PLOT command to plot two variables across time. This ability lets you compare the performance of two variables side-by-side in a single report to see if a cause-and-effect relationship exists between the two. You cannot use the PLOT2 command without the PLOT command.

This command has the following syntax:

```
PLOT2 variable [options][ALONE][SCALE({n},{mX},{COMMON})][CHAR1(x)]
                                                         [CHAR2(x)]
```

The operands for the PLOT2 command are the same as for the PLOT command except for the COMMON option of the SCALE operand. Specify SCALE(COMMON) if you want to use the same scale to plot both variables.

Note: For an explanation of the other PLOT2 operands, see the descriptions of the PLOT operands in PLOT Command (see page 92) in this chapter.

Example: PLOT and PLOT2 Commands

The following PLOT and PLOT2 commands compare the transaction use of transaction CINQ against its lifetime, using the same scale for both variables. Because the ALONE operand is specified, global activity (the values of the variables for all transactions) is not plotted.

```
PLOT  TRAN USE      TRANID(CINQ) ALONE
PLOT2 TRAN LIFETIME TRANID(CINQ) ALONE SCALE(COMMON)
 END
RUN
```

PRTEXIT Command—Define a User Exit

PRTEXIT command defines a user exit to call before each line is printed.

Note: The user exit PRTGUX is predefined for your use.

This command has the following syntax:

```
PRTEXIT(user-exit)
```

user-exit

Defines the name of the user exit.

More information:

User Exits (see page 301)

RANGE Command—Set Criteria to Include

The RANGE command sets criteria that must be met before data is included in a report.

You can use the RANGE command only with Report Writer variables that have numeric values.

This command has the following syntax:

```
RANGE variable [RATE] min1-max1,min2-max2,..[ALONE][DEFERRED][ID rangeid]
                                            [UNDERLINE]
                                            [HIGHLIGHT]
```

variable

Includes any numeric Report Writer variable and then applies the range criteria to that variable.

RATE

Specifies range values as rates (number per second). RATE is valid only when DEFERRED is specified.

min1-max1, min2-max2, ...

Specifies up to 10 minimum and maximum values for the variable chosen. You can specify a single value, or a single value followed by a plus sign (+) to signify no upper boundary. Specifying a pair includes records when the specified variable value is greater than or equal to the minimum value and less than the maximum value.

ALONE

Applies only if multiple RANGE commands are used in a single report. If more than one RANGE command is used, all range criteria must be satisfied before a record is accepted. If one range value fails, the record is discarded.

When you specify the ALONE option on one RANGE command, and the record passes its range criteria:

- The record is accepted
- All other RANGE commands are ignored

DEFERRED

Waits to apply the range criteria until after all the input data is read, and then discards lines having values not meeting the range criteria. For example, only use the DEFERRED operand to report on only those jobs that were executed 10 or more times. The Report Writer is able to determine this criteria only after it reads all the input data. The RATE option is valid for the RANGE command when you use the DEFERRED operand.

UNDERLINE

Like the DEFERRED operand, waits to apply range criteria until after all the input data is read. However, instead of discarding lines not meeting the range criteria, the UNDERLINE option prints all lines and underlines those values that meet a range criterion.

HIGHLIGHT

Similar to UNDERLINE but prints values meeting the range criteria in boldface rather than underlining them. If you print the report on a laser printer, these values do not appear bold. In that case, use the UNDERLINE operand instead of the HIGHLIGHT operand.

ID *rangeid*

In a tabular report, assigns an identifier to the specified range of values for the variable. The range ID must be one to eight characters. You can use the GROUP command to specify a group that includes only those resources that fall within the range named *rangeid.*

Note: For more information, see GROUP Command (see page 72) in this chapter.

Examples: RANGE Command

To plot the activity of transactions with response times under one second, use the following commands:

```
PLOT TRAN USE
RANGE TERM RESP 0-1 DEFERRED
END
RUN
```

The following commands define groups by ranges of transaction lifetime and produce a report showing the average use and lifetime of each group:

```
GROUP TRANID RANGE:00.000-10.000(*) RANGE(TLIFE1)
 GROUP TRANID RANGE:10.001+(*)       RANGE(TLIFE2)
 RANGE TRAN LIFE 0-10     ID TLIFE1
RANGE TRAN LIFE 10-99999 ID TLIFE2
TAB TRANID(RANGE*) TRAN USE AND TRAN LIFE
END
RUN
```

RUN Command—Begin Processing and Printing Reports

The RUN command ends the interpretation of commands and begins the processing and printing of reports.

Make the RUN command the last command in your job stream. Any commands after the RUN command are ignored.

This command has the following syntax:

```
RUN
```

Example: RUN Command

The following is an example of the RUN command in a job stream:

```
TITLE1 REPORT1
PLOT variable
  FROM 02/01/11 00.00.01
  TO   02/15/11 23.59.59
END
*
TITLE1 REPORT2
PLOT variable
  SHIFT 17:00:00 01:00:00
END
RUN
```

SELECT Command—Specify Identifiers

Use the SELECT command to specify the identifiers to include in or excluded from a report.

This command is useful in restricting the following types of reports:

- Sample reports, which cannot otherwise be restricted to specific resources.
- Tabular reports not organized by resource, which can otherwise be restricted only by specifying the ID option and identifiers with every variable included in the report.

This command has the following syntax:

SELECT *idoption*(*identifiers*)

***idoption*(*identifiers*)**

Includes any ID option followed by one or more identifiers of the appropriate type. You can use generic characters when specifying identifiers. Separate multiple identifiers by commas.

Example: SELECT Command

The following commands generate a CICS.TERM.STAT sample report. The SELECT command limits the report to terminals with IDs that do not begin with the letter T.

```
SELECT TERMID(¬T*)
TAB TERMIDS TERMINAL RESP,
        AND TERM MESSAGES,
        AND TERM READS,
        AND TERM WRITES,
        AND TERM INPUTSIZE,
        AND TERM OUTPUTSIZE
END
RUN
```

The following commands generate a tabular report organized by time but not by resource. The SELECT command restricts the report to three transactions, an alternative to specifying the ID option and identifiers TRANID(PAY1,PAY2,PAY3) with each variable.

```
SELECT TRANID(PAY1,PAY2,PAY3)
 TAB TIME TRAN USE                    AND,
         TRAN USE RATE HDR2(RATE)    AND,
         TRAN USE DIFF HDR2(%CHANGE) AND,
         TRAN I/OTIME                AND,
         TRAN MAXI/OTIME             AND,
         TRAN CPUTIME                AND,
         TRAN ABENDS                 AND,
         TRAN GETMSIZE
  EACH HOUR
END
RUN
```

SHIFT Command—Hours Covered for a Report

The SHIFT command restricts the hours of the day covered by a report.

Like most secondary commands, the SHIFT command cannot be used twice for the same report, even if two different periods of time are specified.

The following provides an incorrect use of SHIFT commands in a command stream and the corrected command stream:

- Incorrect Command Stream

```
GRAF variable
   SHIFT 08:00:00 12:00:00
   SHIFT 13:00:00 17:00:00
END
RUN
```

- Corrected Command Stream

```
GRAF variable
   SHIFT 08:00:00 17:00:00

END

GRAF variable
   SHIFT 13:00:00 17:00:00
END
RUN
```

This command has the following syntax:

```
SHIFT start-time end-time
```

start-time

Specifies the earliest time to include in the report, in *hh:mm:ss* format, where *hh* is hours, *mm* is minutes, and *ss* is seconds. You can use periods instead of colons to separate the hours, minutes, and seconds.

end-time

Specifies the latest time to include in the report, in the same format as *start-time.*

Examples: SHIFT Command

To graph only the activity that occurs between 8:00 a.m. and 5:00 p.m., enter the following:

```
GRAF variable
  SHIFT 08:00:00 17:00:00
END
RUN
```

Shifts can cross midnight. For example, to write a tabular report that covers from 11:00 p.m. to 7:00 a.m., use the following commands:

```
TAB variable
  SHIFT 23:00:00 07:00:00
END
RUN
```

TABulate Command—Produce Tabular Reports

The TAB command produces tabular reports. The variables you specify with the TAB command determine the columns in your report and the order in which they appear.

The operands you specify determine:

- How the report is organized: by time, by classes of or individual *identifiers* (such as transactions, terminals, or users), or by the relationship between identifiers
- How the report is sorted
- The column headings that appear in the report

This command has the following syntax:

```
TAB [period] [idoption[(identifiers)]] variable [options]
 [HDR1(text)] [HDR2(text)] [SORT order] [FOR n] [AND,
 [TOP n]
 [BOTTOM n]
 variable [options] [HDR1(text)] [HDR2(text)]] [AND,...]
```

period

Sets the first column in the report to one of the following:

- DATETIME - Display the date and time
- DATE - Display the date
- TIME - Display the time
- DAY - Display the day (MONDAY, TUESDAY, and so on)

***idoption*(*identifiers*)**

Includes any ID option that is valid for the variable, specified with or without identifiers. Specifying this operand organizes the resulting report by the specified ID option and limited to the identifiers specified by *identifiers.* You can specify multiple ID options.

The identifiers are listed either:

- Beginning in the first column (if you did not specify a time period as the first column) or
- Beginning in the second column

variable

Includes any Report Writer variable.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

HDR1(*text*)

Overrides the default top heading for a column. Each column has two default eight-character lines. Replace *text* with the new column heading. To remove the header, replace *text* with a null string, as in HDR1( ).

HDR2(*text*)

Overrides the default bottom heading for a column. Replace *text* with the new column heading. To remove the header, replace *text* with a null string, as in HDR2( ).

SORT *order*

Sorts the report. Replace *order* with one of the following:

- (Default) DESCENDING

  Sorts variables in descending order. The order starts with the variable specified with the SORT operand, and, when the value is the same for two entries, from left to right.

- ASCENDING

  Sorts variables in ascending order. The order starts with the variable specified with the SORT operand, and, when the value is the same for two entries, from left to right.

- ALPHA

  Sorts variables in alphanumeric order with respect to the specified identifier; for example, by transaction or terminal.

You can use SORT only if you organize the report by identifier; for example, by transaction. If you organize the report by time, then it is already sorted.

FOR *n*

For Reports Organized by Identifier and Time or Date: Limits your reports to the *n* identifiers in each time period that would appear first in the report, where *n* is an integer. For reports sorted in descending order, these reports are the *n* most active identifiers. For reports sorted in ascending order, these reports are the *n* least active identifiers.

For Reports Organized by Time or Date: Limits your reports to the *n* times or dates with the highest values for the first variable specified. For reports sorted by a variable in descending order, these reports are the *n* times or dates with the highest value for that variable. For reports sorted by a variable in ascending order, these reports are the *n* times or dates with the lowest value for that variable.

TOP *n*

Limits reports organized by identifier to the *n* identifiers with the highest values for the first variable specified, where *n* is an integer. Unlike the FOR operand, the TOP operand tests each record as it is read and discards records that do not qualify, saving storage.

The TOP operand is appropriate for ID options with many unique identifiers, such as TRANNUM.

BOTTOM *n*

Identical to the TOP operand except it prints the *n* identifiers with the lowest values for the first variable specified.

AND,

Signals that another variable is being specified. The comma after AND lets you continue to the next line.

Examples: TAB Command

Create a tabular report organized by date and time that shows the transaction use and average transaction lifetime for all transactions each hour:

```
TAB DATETIME TRAN USE    AND,
             TRAN LIFE
   EACH HOUR
END
RUN
```

Create a tabular report organized by date and time that shows the five hours with the highest average transaction lifetime:

```
TAB DATETIME TRANID TRAN LIFE FOR 5
  EACH HOUR
END
RUN
```

More information:

Tabular Reports (see page 191)

TAB2 Command—Build a Secondary Table

The TAB2 command builds a secondary table for each row of the table created by the TAB command.

This command has the following syntax:

```
TAB2 idoption[(identifiers)] variable [options] [HDR1(text)] [HDR2(text)]
[SORT order] [FOR n] [AND,
 [TOP n]
 [BOTTOM n]
 variable [options] [HDR1(text)] [HDR2(text)]] [AND,...]
```

The TAB2 command uses the same operands as the TAB command, except that you cannot specify both a period operand and an ID option. The period operands are DATETIME, DATE, TIME, or DAY.

Note: For an explanation of the TAB2 operands, see the description of the TAB operands in TABulate Command (see page 100) in this chapter.

You cannot use the TAB2 command without the TAB command.

You can use the TAB2 command only if you also specify the TAB command with an ID option. You can specify a period option (DATETIME, DATE, TIME, and DAY) with either the TAB command or the TAB2 command, but *not* with both.

Examples: TAB2 Command

The following commands create a tabular report organized first by transaction ID and then by the date:

```
TAB  TRANID TRAN USE
TAB2 DATE   TRAN USE
    EACH HOUR
END
RUN
```

More information:

Tabular Reports (see page 191)

TITLE1, TITLE2, and TITLE3 Commands

Three 60-character lines at the top of each page are reserved for titles. The TITLE1, TITLE2, and TITLE3 commands write titles on the first, second, and third of these lines, respectively. If you do not use the TITLE commands, the Report Writer assigns its own title lines to the report.

Titles are printed only for the report in which the TITLE commands occur. Use the GLOBAL *TITLEn* command to print the same title for all reports in one run.

The TITLE1 title is printed first, the TITLE2 title is printed below it, and the TITLE3 title appears last. Specify the TITLE1, TITLE2, and TITLE3 commands in that order. You can use none, one, or all of these commands in any combination.

These commands have the following syntax:

```
TITLE1 titletext
TITLE2 titletext
TITLE3 titletext
```

titletext

Specifies the title that you assign to the report, up to 60 characters. Every character in the title text must be on the same line as the TITLE1, TITLE2, or TITLE3 command. You cannot continue titles with a continuation character. To insert dynamically dates and times into *titletext*, use the keywords described in the next section.

TITLE*n* has the following keywords:

You can use keywords to place dynamically dates and times in your report titles. Keywords have the following format:

```
&SourceRangeFormat
```

Replace the variables as shown:

&Source

Replace with one of the following to determine the origin of the dates and times:

C-Uses report command specifications

R-Uses the starting and ending times for the data in the report

E-Uses the EACH command specifications

P-Uses the PERIOD command specifications

Range

Replace with one of the following to determine whether to take data from the FROM command or the TO command:

F-Uses the FROM command specification

T-Uses the TO command specification

Format

Replace with one of the following to determine the format of the date and time:

WEEKDAY = Monday, Tuesday, and so on

DATE = *mm/dd/yy* or *dd/mm/yy* (European format)

DAY = *dd*

EDAY = *dd*, if your dates are in European format

MONTH = *mm*

EMONTH = *mm*, if your dates are in European format

YEAR = *yy*

TIME = *hh:mm:ss*

HOUR = *hh*

MINUTE = *mm*

SECOND = ss

CALMONTH = January, February, and so on

JDAY = *dd* (Julian date)

Note: Do not use the &RF and &RT variables with the FLASHBACK reports. The way the FLASHBACK reports process records prevents enumerating the date and time range of the records.

Examples: TITLE1, TITLE2, TITLE3 Commands

- To have the title *Transaction Use by Hour* appear on title line 2, enter the following command:

  ```
  TITLE2 Transaction Use by Hour
  ```

- The following commands dynamically create a date and time on title lines 2 and 3. The data in the report and the FROM and TO commands determine the date and time:

  ```
  TITLE2 FROM: &RFWEEKDAY, &RFCALMONTH &RFDAY, 20&RFYEAR at &RFTIME
  TITLE3 TO: &RTWEEKDAY, &RTCALMONTH &RTDAY, 20&RTYEAR at &RTTIME
  ```

- The following is sample output for these commands:

  ```
  FROM: Thursday, January 27 2011 at 02.58.00
  TO: Tuesday, February 01 2011 at 11.23.59
  ```

VPLOT Command—Plot Usage in Vertical Columns

The VPLOT command plots resource usage in vertical columns. The horizontal axis of the plot represents time.

This command has the following syntax:

```
VPLOT variable [options] [SCALE({n},{m}X)] [STACK=id] [CHAR1(x)]
      [HDR1(text)] [HDR2(text)] [HDR(text)]
```

variable

Indicates any Report Writer variable.

options

Specifies any valid variable options or ID options. Each ID option you specify must include at least one identifier.

SCALE(*n*) or SCALE(*m*X) or SCALE(*n*,*m*X)

Overrides the default maximum plot scale.

By default, the highest value of the vertical axis of a plot equals the highest value reached by a variable in the plot. Or, for variables representing percentages, 100 percent.

To override this dynamic scale, specify *n* as the maximum value. If the variable that you use produces integers, *n* must be an integer.

You can also widen each plot column by using the multiplier *m*X, where *m* is an integer.

STACK=*id*

Replaces the plot character with the identifiers being plotted, from most active to least active starting at the bottom of the plot. You can specify the *m*X option of the SCALE operand to print more characters of each identifier.

Note: For an example using the STACK operand, see the VPLOT command in the chapter “Plot Reports.”

CHAR1(*x*)

Specifies the plot character. Replace *x* with any valid EBCDIC character.

HDR1(*text*)

Overrides the default top heading for the scale to the left of the plot. Replace *text* with the new heading. To remove the header, replace *text* with a null string, as in HDR1( ).

HDR2(*text*)

Overrides the default bottom heading for the scale to the left of the plot. Replace *text* with the new heading. To remove the header, replace *text* with a null string, as in HDR2( ).

HDR(*text*)

Overrides the default description next to the SYMBOLS heading that describes the plotted variable. Replace *text* with the new description, not to exceed 30 characters.

Example: VPLOT Command

The following commands plot transaction use against time using the letter T as the plot character:

```
VPLOT TRANSACTION USE CHAR1(T)
END
RUN
```

VPLOT2 Command—Plot Two Variables

The VPLOT2 command is used with the VPLOT command to plot two variables across time. This ability lets you compare the performance of two variables side-by-side in a single report to see if a cause-and-effect relationship exists between them. The VPLOT2 variable is printed directly over the VPLOT variable.

Note: Use the VPLOT2 command with the VPLOT command.

This command has the following syntax:

```
VPLOT2 variable [options] [SCALE({n},{mX},{COMMON})] [CHAR1(x)]
 [HDR1(text)] [HDR2(text)] [HDR(text)]
```

The operands for the VPLOT2 command are the same as for the VPLOT command except that the VPLOT2 command:

- Does not support the STACK operand
- Does support the COMMON option of the SCALE operand

Specify SCALE(COMMON) if you want to use the same scale to plot both variables.

Example: VPLOT and VPLOT2 Commands

The following commands plot terminal response time with asterisks (*) and transaction use rate with plus signs (+):

```
VPLOT  TERM RESP             CHAR1(*)
VPLOT2 TRANSACTION USE RATE  CHAR1(+)
END
RUN
```

More information:

VPLOT Command—Plot Usage in Vertical Columns (see page 106)

Chapter 5: Variables

This section contains the following topics:

Functions of Variables and Options (see page 109)
Conventions Used for Describing Variables (see page 111)
Source for MVS Variable Values (see page 112)
CICS Variables (see page 112)
IMS Variables (see page 129)
MQ Variables (see page 132)
AUDIT Variables (see page 137)
COMMON Variables (see page 138)
MVS Variables (see page 140)

Functions of Variables and Options

The variables you use with a report command determine what the report describes. For example, if you use the command GRAF TERMINAL USE (where TERMINAL USE is the variable), you get a report that graphs terminal use.

You can specify the following two types of options with variables:

- Variable options, which change either the type of data that a variable represents or its format.

  For example, you can change:

  - The type of data from a count to a rate
  - The format from one decimal place to three
- ID options, which, when used with variables, change the scope of the data that the variables represent. For example, the scope of the data can be changed from transaction use for all transactions to the transaction use for transaction CINQ only.

More information:

ID Options (see page 167)
Variable Options (see page 157)

Structure and Parts of Variable Names

Report variables have the following format:

`[system] [record-type] variable-class variable-name`

system

Specifies the system of the variable. Valid values are as follows:

CICS

Signals to the Report Writer that the variable is a CA SYSVIEW-based CICS variable.

IMS

Signals that the variable is a CA SYSVIEW-based IMS variable.

MQ

Signals that the variable is a CA SYSVIEW-based MQ variable.

AUDIT

Signals that the variable is a CA SYSVIEW-based AUDIT variable.

COMMON

Signals that the variable is a CA SYSVIEW-based variable common for all systems.

MVS

Signals to the Report Writer that the variable is a z/OS system variable.

Default: CICS

record-type

Specifies the type of record the data for that variable comes from. You can specify one of the following:

JOB

Provides job activity information.

PERformance

Provides task activity information.

REQuest

Provides information about MQ objects and message counts.

STEP

Provides activity information for the last interval before the step terminate, and activity after the previous interval ended.

SUMmary

For CICS variables, provides information from data created with the SUMMARIZE utility option.

For IMS variables, provides information from IMS region summary records.

SYStem

For CICS variables, provides information from system threshold records.

For MVS variables, provides information from MVS system records.

INTerval

Provides CICS resource usage information. The collection of this data is based upon user-defined time intervals and involves low overhead.

LOG

Provides CA SYSVIEW Audit Event records information.

EXCEPTION

Provides CA SYSVIEW exception records information.

The default record type for all CICS and IMS variables is PERformance and for all MVS variables is SYStem. The record type that a variable represents affects which ID options can be used with it to limit the scope of a report.

Note: For more information about JOB and STEP record types, see the section JOB and STEP Record Type Data.

variable-class

Specifies one of the variable classes. For example, TRANSACTION and DATASET are both valid variable classes for the PERformance record type.

variable-name

Specifies a variable name from the variable class selected. For example, both USE and LIFETIME are valid variable names for the variable class TRANSACTION.

Conventions Used for Describing Variables

The following conventions are used for describing variables:

- All variables that represent time are displayed in seconds.
- Variable names separated by a vertical bar (|) are synonymous.
- Default values are underlined.
- Characters in a variable class or name that you must enter are shown in capital letters. Optional characters are shown in lowercase letters.

Source for MVS Variable Values

The tables describing MVS variables contain a Data column, which gives the SMF record from which the variable obtains its data.

JOB and STEP Record Type Data

Variable data for all MVS variables with record type JOB or STEP is accumulated from SMF30 type records, as follows:

- Variables with record type JOB use subtype 5 records. Subtype 5 records are created at the end of the job and represent the entire activity for that job.
- Variables with record type STEP use subtype 2 and subtype 3 records, which are delta-count records. Subtype 2 records are created at each SMF interval. Subtype 3 records are created at the end of each step.

SMF dsect Field Names Used When Specifying Variables

All SMF30 DSECT field names that refer to character data are supported as valid identifiers. When generating reports, you can use these SMF30 DSECT field names in place of the variable name part of the complete variable.

For example, the first column of the following table lists variables specified using variable names. The second column lists the same variables specified using the SMF30 DSECT field name in place of the variable name part of the complete variable.

Variable Specified Using Variable Name	Same Variable Specified Using the SMF DSECT Field Name
MVS JOB STORage BELOW	MVS JOB STORage SMF3OURB
MVS STEP PERF TOT-SU	MVS STEP PERF SMF30RSV
MVS JOB EXCP COUNT	MVS JOB EXCP SMF30BLK
MVS STEP EXCP DEV-CONN	MVS STEP EXCP SMF30DCT

CICS Variables

The following pages describe the CICS variables and the variable names for each class defined.

CICS CDSA Variable Class

This section describes variable information for the CDSA class on the CICS system.

Syntax:

`[CICS] [record-type] CDSA variable name`

ID Options

Use any CICS CDSA variables with the following ID options: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	AVAILABLE	23, 28	Storage available, in bytes
INTerval	Size	23, 28	Storage amount, in bytes
INTerval	%USE	23, 28	Percentage of storage in use

CICS DATASET Variable Class

This section describes variable information for class DATASET on system CICS.

Syntax:

`[CICS] [record-type] DATASET variable-name`

ID Options

Use the CICS Performance data set variables with the following ID options: ABCODE, APPLID, FILEID, JOBNAME, PROGID, SYSID, TERMID, TRANID, TRANNUM, USERID, and VTAMLU.

Note: You can specify FILE in place of DATASET for this variable class.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	ADDs	27	Number of files added
PERFormance	BROWSEs	27	Number of file browses
PERFormance	DELetes	27	Number of files deleted
PERFormance	MAXREQtime	27	Maximum file request service time, in seconds

Record Type	Variable Name	SMF Subtype	Description
PERFormance	READs	27	Number of read-updates, read no updates, and browses
PERFormance	READNUpdts	27	Number of files read no update
PERFormance	READUpdts or GETU	27	Number of files read for update
PERFormance	REQTIME	27	Average file request service time, in seconds
PERFormance	UPDATEs	27	Number of file updates
PERFormance	USE, USAGE, RW, or REQUESTS	27	Number of file requests
PERFormance	WRITEs	27	Number of updates, adds, and deletes

CICS DB2 Variable Class

This section describes variable information for class DB2 on system CICS.

Syntax:

```
CICS PERFormance DB2 variable-name
```

ID Options:

Use CICS PERFormance DB2 variables with the following ID options: ABCODE, APPLID, DB2PROG, DB2 STYPE, DB2PSNUM, JOBNAME, SYSID, TERMID, TRANID, TRANNUM, USERID and VTAMLU.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	REQUEST	27	Number of DB2 requests
PERFormance	REQTime	27	Average DB2 request time, in seconds
PERFormance	MAXREQTime	27	Maximum DB2 request time, in seconds

CICS DLI Variable Class

This section describes variable information for class DLI on system CICS.

Syntax:

`[CICS] [record-type] DLI variable-name`

ID Options

Use CICS PERFormance DLI variables with the following ID options: ABCODE, APPLID, FILEID, JOBNAME, SYSID, TERMID, TRANID, TRANNUM, USERID, and VTAMLU

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	DELETEs	27	Number of DL/I delete requests
PERFormance	GETNEXT or GN	27	Number of DL/I GET next requests
PERFormance	GETUNIQUE or GU	27	Number of DL/I GET unique requests
PERFormance	GHN	27	Number of DL/I GET hold next requests
PERFormance	GHNP	27	Number of DL/I GET hold next within parent requests
PERFormance	GHU	27	Number of DL/I GET hold unique requests
PERFormance	GNP	27	Number of DL/I GET next within parent requests
PERFormance	INSERTs	27	Number of DL/I insert requests
PERFormance	READs	27	Number of DL/I read requests, including GETNEXT, GNP, GU, GHU, GHN, and GHNP
PERFormance	REPLACEs	27	Number of DL/I replace requests
PERFormance	SERVTIME	27	Average DL/I replace requests
PERFormance	USE, USAGE, REQUESTs, or IO	27	Number of DL/I file requests
PERFormance	WRITEs	27	Number of DL/I write requests, including replaces, deletes, and inserts

CICS ECDSA Variable Class

This section describes variable information for class ECDSA on system CICS.

Syntax:

`[CICS] [record-type] ECDSA variable-name`

ID Options

Use any CICS ECDSA variable with the following ID options: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

Information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	AVAILABLE	23, 28	Storage available, in bytes
INTerval	Size	23, 28	Storage amount, in bytes
INTerval	%USE	23, 28	Percentage of storage in use

CICS ERDSA Variable Class

This section describes variable information for class ERDSA on system CICS.

Syntax:

`[CICS] [record-type] ERDSA variable-name`

ID Options

Use any CICS ERDSA variable with the following ID options: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	AVAILABLE	23, 28	Storage available, in bytes
INTerval	Size	23, 28	Storage amount, in bytes
INTerval	%USE	23, 28	Percentage of storage in use

CICS EUDSA Variable Class

This section describes variable information for class EUDSA on system CICS.

Syntax:

`[CICS] [record-type] EUDSA variable-name`

ID Options

Use any CICS EUDSA variable with the following ID options: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	AVAILABLE	23, 28	Storage available, in bytes
INTerval	Size	23, 28	Storage amount, in bytes
INTerval	%USE	23, 28	Percentage of storage in use

CICS EXCEPTION Variable Class

This section describes variable information for class EXCEPTION on system CICS.

Syntax:

`[CICS] [record-type] EXCEPTION variable-name`

ID Options

The record type of a CICS EXCEPTION variable determines which ID options can be used with the variables.

- Use any CICS PERFormance EXCEPTION variables with the following ID options:

  ABCODE, APPLID, CICSID, JOBID, NETNAME, REGID, SYSID, TERMID, TRANNUM, USERID, and VTAMLU

- Use any CICS SYStem EXCEPTION variable with the following ID options:

  CICSREL, JOBNAME, SYSID, TRANID, TERMID, USERID, and VTAMLU

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance, SYStem	EXCMNEXN	27, 24	Exception number

Record Type	Variable Name	SMF Subtype	Description
PERFormance, SYStem	EXCMNLUN	27, 24	LU name
PERFormance, SYStem	EXCMNRID	27, 24	Exception resource ID
PERFormance, SYStem	EXCMNRTY	27, 24	Exception resource type
PERFormance, SYStem	EXCMNSTA	27, 24	Start date and time of exception
PERFormance, SYStem	EXCMNSTO	27, 24	Stop date and time of exception
PERFormance, SYStem	EXCMNTER	27, 24	Terminal ID
PERFormance, SYStem	EXCMNTNO	27, 24	Transaction number
PERFormance, SYStem	EXCMNTPR	27, 24	Transaction priority
PERFormance	EXCMNTRN	27, 24	Transaction ID
PERFormance, SYStem	EXCMNTST	27, 24	Transaction start type
PERFormance, SYStem	EXCMNTYP	27, 24	Exception type
PERFormance, SYStem	EXCMNUSR	27, 24	User ID

CICS PAGE Variable Class

This section describes variable information for class PAGE on system CICS.

Syntax:

[CICS] [*record-type*] PAGE *variable-name*

ID Options

Use CICS INTERVAL (INT) PAGE variables with the following ID option:

CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	CACHE-MISS	23, 28	Number of cache read page misses
INTerval	IN	23, 28	Number of page-ins
INTerval	IN-COMMON	23, 28	Number of common page-ins
INTerval	IN-HYPERSPACE	23, 28	Number of hyperspace page-ins
INTerval	IN-LPA	23, 28	Number of LPA page-ins

Record Type	Variable Name	SMF Subtype	Description
INTerval	IN-NONVAM	23, 28	Number of non-VAM page-ins
INTerval	IN-SWAP	23, 28	Number of pages swapped in
INTerval	IN-VAM	23, 28	Number of VAM page-ins
INTerval	OUTs	23, 28	Number of page-outs
INTerval	OUT-HYPERSPACE	23, 28	Number of hyperspace page-outs
INTerval	OUT-NONVAM	23, 28	Number of non-VAM page-outs
INTerval	OUT-SWAP	23, 28	Number of pages swapped out
INTerval	RECLAIMs	23, 28	Number of pages reclaimed
INTerval	STEALs	23, 28	Numbers of pages stolen
INTerval	SWAPs	23, 28	Number of swaps that occurred

CICS PROGRAM Variable Class

This section describes variable information for class PROGRAM on system CICS.

Syntax:

`[CICS] [record-type] PROGRAM variable-name`

ID Options

Use only PERFormance PROGram variables with the following ID options: ABCODE, APPLID, JOBNAME, PROGID, SYSID, TERMID, TRANID, TRANNUM, USERID, and VTAMLU.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	RESPonsetime	27	Average program response time, in seconds
PERFormance	USE	27	Number of programs used

CICS REGION Variable Class

This section describes variable information for class REGION on system CICS.

Syntax:

`[CICS] [record-type] REGION variable-name`

ID Options

Use any INTerval REGION variable with the following ID options: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	CPUTIME	23, 28	Total CPU time consumed by the CICS region
INTerval	SIOs	23, 28	Number of SIOs executed by the CICS region
INTerval	THRESHOLDs	23, 28	Number of thresholds triggered in the CICS region

CICS SYSTEM Variable Class

This section describes variable information for class SYSTEM on system CICS.

Syntax:

`[CICS] [record-type] SYSTEM variable-name`

ID Options

Use any CICS SYSTEM variable with the following ID options: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	AVAILABLE	23, 28	Available frame count
INTerval	CPU	23, 28	CPU usage
INTerval	PAGING	23, 28	Demand page rate
INTerval	UIC	23, 28	Unreferenced interval count

CICS TEMPSTOR (Temporary Storage) Variable Class

This section describes variable information for class TEMPSTOR on system CICS.

Syntax:

`[CICS] [record-type] TEMPSTOR variable-name`

ID Options

Use only PERFormance TEMPSTOR variables with the following ID options: ABCODE, APPLID, JOBNAME, SYSID, TERMID, TRANID, TRANNUM, TSTGID, USERID, and VTAMLU.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	BGET	27	Average number of temporary storage bytes per GET
PERFormance	BREQ	27	Average number of temporary storage bytes per request
PERFormance	READs	27	Number of GET requests for temporary storage
PERFormance	GETMAXRESPonse	27	Maximum temporary storage GET response time
PERFormance	GETMAXSIZE	27	Maximum temporary storage GET request size, in bytes
PERFormance	GETRESPonse	27	Average temporary storage GET response time per request
PERFormance	MAXRESP	27	Maximum temporary storage request time, in seconds
PERFormance	PUTMAXRESPonse	27	Maximum temporary storage PUT response time
PERFormance	PUTs	27	Number of temporary storage PUT requests
PERFormance	REQUESTs	27	Number of temporary storage requests
PERFormance	RESPONSE	27	Average temporary storage request time, in seconds, per request

CICS TERMINAL Variable Class

This section describes variable information for class TERMINAL on system CICS.

Syntax:

`[CICS] [record-type] TERMINAL variable-name`

ID Options

Use only performance terminal variables used with ID options. All ID options are valid except DSAID, and LSRID.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	INPUTS	27	Number of terminal input messages
PERFormance	INPUTSIZE	27	Average input message size
PERFormance	INPUTSIZE-LU6.1	27	Average LUTYPE6.1 input message size
PERFormance	INPUT-LU6.1	27	Number of LUTYPE6.1 alternate terminal input messages
PERFormance	MESSAGES	27	Number of messages
PERFormance	MROWATtime	27	Average MRO I/O wait time per transaction, in seconds
PERFormance	MSGSIZE	27	Average message size in bytes
PERFormance	OUTPUTS	27	Number of terminal output messages
PERFormance	OUTPUTSIZE	27	Average output message size
PERFormance	OUTPUT-LU6.1	27	Number of LUTYPE6.1 alternate terminal output messages
PERFormance	OUTPUTSIZE-LU6.1	27	Average LUTYPE6.1 output message size
PERFormance	RESPonsetime	27	Average terminal response time, in seconds
PERFormance	TCTTE-REQuests	27	Number of TCTTE ALLOCATE requests for APPC, LUTYPE6.1 and IRC sessions
PERFormance	USE, USAGE, or WRITES	27	Number of writes to terminal

CICS THRESHOLD Variable Class

This section describes variable information for class THRESHOLD on system CICS.

Syntax:

`[CICS] [`*record-type*`] THRESHOLD` *variable-name*

ID Options

Use only PERFormance THRESHOLD variables with the following ID options. The following ID options are valid: ABCODE, APPLID, FILEID, JOBNAME, PROGID, SYSID, TERMID, THRCLAS, THRSID, THRSSID, THRTYPE, TRANID, TRANNUM, TRSCEID, USERID, and VTAMLU.

Use only SYStem THRESHOLD variable with the following ID options: JOBNAME, SYSID, THRCLAS, THRSID, THRSSID, THRTYPE, TRANID, TRANNUM, TRSCEID, and USERID.

The following table provides variable information for the CICS system with a variable class of threshold:

Record Type	Variable Name	SMF Subtype	Description
PERFormance, SYStem	CLASS	27, 08	COUNT, RATE, TIME, and PCT threshold
PERFormance, SYStem	COUNT	27, 08	Number of thresholds exceeded
SYStem	DATE	08	Date threshold occurred on *mm*/*dd*/*yy*.
PERFormance, SYStem	LIMIT	27, 08	Threshold limit value
PERFormance, SYStem	MAXVALUE	27, 08	Maximum value that exceeded the limit
PERFormance, SYStem	MINVALUE	27, 08	Minimum value that exceeded the limit
SYStem	NAME	08	Threshold name
PERFormance, SYStem	RESOURCE	27, 08	Resource involved with the threshold
SYStem	TERMINAL	08	Terminal ID, if task related
SYStem	TIME	08	Time threshold occurred at *hh*:*mm*:*ss*
SYStem	TRANSACTION	08	Transaction ID, if task related
PERFormance, SYStem	TYPE	27, 08	UPPER or LOWER level threshold
SYStem	USER	08	User ID, if task related
PERFormance, SYStem	VALUE	27, 08	Average value that exceeded the limit

CICS Interval Record TRANsaction Variable Class

This section describes variable information for class TRANsaction on system CICS.

Syntax:

```
[CICS] [record-type] TRANSACTION variable-name
```

ID Options

Use any interval transaction variable with the following ID options: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

The following table provides variable information for the CICS system with a variable class of transaction:

Record Type	Variable Name	SMF Subtype	Description
INTerval	CPUtime	23, 28	Average CPU time, in seconds
INTerval	I/Otime	23, 28	Average time, in seconds, spent waiting for I/O
INTerval	LIFEtime	23, 28	Average transaction lifetime, in seconds
INTerval	SUSPtime	23, 28	Average time, in seconds, spent suspended
INTerval	TCPUtime	23, 28	Total CPU time for the transaction including user, CICS, and CA SYSVIEW time
INTerval	TMIOtime	23, 28	Average time, in seconds, spent waiting for terminal
INTerval	USE	23, 28	Number of transactions executed
INTerval	WTRtime	23, 28	Average time, in seconds, spent waiting to run

CICS Performance Record TRANsaction Variable Class

This section describes variable information for class TRANsaction on system CICS.

Syntax:

```
[CICS] [record-type] TRANSACTION variable-name
```

ID Options

The record type of a TRANsaction variable determines which ID options can be used with the variable, as follows:

- Use PERFormance TRANsaction variables with all ID options except DSAID, and LSRID.
- Use any CICS SUMmary TRANsaction variables with the following ID options: CICSREL, JOBNAME, SYSID, and TRANID.

The following table provides variable information for the CICS system with a variable class of transaction:

Record Type	Variable Name	SMF Subtype	Description
PERFormance, SUMmary	ABENDS	27, 25	Number of abends
PERFormance	BMS-MAP	27	Number of map requests
PERFormance	BMS-MAPIN	27	Number of map-in requests
PERFormance	BMS-MAPOUT	27	Number of map-out requests
PERFormance	BMS-REQuests	27	Total number of BMS requests
PERFormance, SUMmary	CPUtime	27, 25	Average CPU time, in seconds
PERFormance	DCOUNT	27	Average number of transient data sets used
PERFormance	ENDDATE	27	Transaction end date
PERFormance	ENDTIME	27	Transaction end time
PERFormance	EXCEPTIONtime	27	Average time spent waiting on exception conditions, in seconds
PERFormance	FCOUNT	27	Average number of files accessed
PERFormance	FC-AC	27	Number of file access method requests
PERFormance	FC-ADD	27	Number of add file requests
PERFormance	FC-BROWSE	27	Number of file browse requests
PERFormance	FC-DELETE	27	Number of file delete requests
PERFormance	FC-GET	27	Number of GET file requests
PERFormance	FC-IOWAITtime	27	Average file I/O wait time
PERFormance	FC-PUT	27	Number of file PUT requests
PERFormance	FC-REQuests	27	Number of file requests

Record Type	Variable Name	SMF Subtype	Description
PERFormance	IRIOtime	27	Average time spent waiting on MRO links, in seconds
PERFormance, SUMmary	I/Otime	27, 25	Average time, in seconds, spent waiting for I/O
PERFormance	JCIOtime	27	Average time spent waiting on Journal I/O, in seconds
PERFormance, SUMmary	LIFEtime	27, 25	Average transaction lifetime, in seconds
PERFormance	MAXI/Otime	27	Maximum time, in seconds, spent waiting for I/O
PERFormance	MAXLIFE	27	Maximum transaction lifetime, in seconds
PERFormance	MAXPGMtime	27	Maximum time, in seconds, spent in program code, including both user and CICS code
PERFormance	MAXSUSPtime	27	Maximum time, in seconds, spent suspended
PERFormance	MAXTMIOtime	27	Maximum time, in seconds, spent waiting for terminal
PERFormance	MAXWAITtime	27	Maximum time, in seconds, spent waiting for an event control block
PERFormance	PCIOtime	27	Average time spent waiting on program loads, in seconds
PERFormance	PCOUNT	27	Average number of programs used
PERFormance	PC-LINKs	27	Number of program links
PERFormance	PC-LOADs	27	Number of program loads
PERFormance	PC-LOADTIME	27	Average program load time
PERFormance	PC-XCTLs	27	Number of program XCTLs
PERFormance	PGMtime	27	Average time, in seconds, spent in program code, including both user and CICS code
PERFormance	SC-GETMABOVE	27	Average number of user storage GETMAIN requests above the 16M line per transaction
PERFormance	SC-GETMBELOW	27	Average number of user storage GETMAIN requests below the 16M line per transaction

Record Type	Variable Name	SMF Subtype	Description
PERFormance	SC-PGMHWM	27	Maximum amount of program storage in use
PERFormance	SC-PGMHWMBELOW	27	Maximum amount of program storage in use below the 16-MB line
PERFormance	SC-STGOCCABOVE	27	Average storage occupancy above the 16-MB line per transaction, in byte-milliseconds (a measurement of storage in use against elapsed time)
PERFormance	SC-STGOCCBELOW	27	Average storage occupancy below the 16-MB line per transaction, in byte-milliseconds (a measurement of storage in use against elapsed time)
PERFormance	SC-TERMSTG	27	Average amount of terminal storage (TIOA) allocated to the associated terminal
PERFormance	SC-USERHWMABOVE	27	Maximum amount of user storage allocated above the 16-MB line
PERFormance	SC-USERHWMBELOW	27	Maximum amount of user storage allocated above the 16-MB line
PERFormance	SP-SYNCPOINTs	27	Number of SP sync point requests
PERFormance	STARTDATE	27	Transaction start date
PERFormance	STARTTIME	27	Transaction start time
PERFormance, SUMmary	SUSPtime	27, 25	Average time, in seconds, spent suspended
PERFormance, SUMmary	TCPUtime	27, 25	Total CPU time for the transaction including user, CICS, and CA SYSVIEW time
PERFormance	TD-GET	27	Number of transient data GET requests
PERFormance	TD-IOWAITtime	27	Average transient data I/O wait time
PERFormance	TD-PURGE	27	Number of transient data purge requests
PERFormance	TD-PUT	27	Number of transient data PUT requests
PERFormance	TD-REQuest	27	Number of transient data requests
PERFormance	TMIOtime	27	Average time, in seconds, spent waiting for terminal
PERFormance	TS-GETS	27	Number temporary storage GET requests
PERFormance	TS-IOWAITtime	27	Average temporary storage I/O wait time per transaction

Record Type	Variable Name	SMF Subtype	Description
PERFormance	TS-PUTAux	27	Number temporary storage PUT auxiliary requests
PERFormance	TS-PUTMain	27	Number temporary storage PUT main requests
PERFormance	TS-REQuests	27	Number temporary storage requests
PERFormance, SUMmary	USE|USAGE	27, 25	Number of transactions executed
PERFormance, SUMmary	WAITtime	27, 25	Average time, in seconds, spent waiting for an event control block

CICS TRANSIENT Variable Class

This section describes variable information for class TRANSIENT on system CICS.

Syntax:

`[CICS] [record-type] TRANSIENT variable-name`

ID Options

The following ID options can be used with PERFormance TRANSIENT variables: ABCODE, APPLID, DCTID, JOBNAME, SYSID, TERMID, TRANID, TRANNUM, USERID, and VTAMLU.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	INPUTMAXRESPonse	27	Maximum transient data input response time, in seconds
PERFormance	INPUTMAXSIZE	27	Maximum transient data input request size, in bytes
PERFormance	INPUTREponse	27	Average transient data input request size, in seconds
PERFormance	INPUTSIZE	27	Average transient data input request size, in bytes
PERFormance	OUTPUTMAXRESPonse	27	Maximum transient data output response time, in seconds
PERFormance	OUTPUTMAXSIZE	27	Maximum transient data output request size, in bytes

Record Type	Variable Name	SMF Subtype	Description
PERFormance	OUTPUTRESPonse	27	Maximum transient data output response time per request, in seconds
PERFormance	OUTPUTSIZE	27	Average transient data output request size, in bytes
PERFormance	READs|INPUTs	27	Number of reads of transient data sets
PERFormance	REQUESTs	27	Number of I/Os to transient data sets
PERFormance	WRITEs|OUTPUTs	27	Number of writes to transient data sets

CICS UDSA Variable Class

This section describes variable information for class UDSA on system CICS.

Syntax:

`[CICS] [`*`record-type`*`] UDSA `*`variable-name`*

ID Options

The following ID options can be used with any CICS UDSA variable: CICSREL, JOBNAME, STARTTIME, ENDTIME, and SYSID.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
INTerval	AVAILABLE	23, 28	Storage available, in bytes
INTerval	Size	23, 28	Storage amount, in bytes
INTerval	%USE	23, 28	Percentage of storage in use

IMS Variables

This section describes the IMS variables and the variable names for each class defined.

IMS TRANsaction Variable Class

This section describes variable information for class TRANsaction on system IMS.

Syntax:

`[IMS] [record-type] TRANSACTION variable-name`

ID Options

The following ID options can be used with any IMS TRANSACTION variable: IMSID, IMSREL, JOBNAME, LTERM, LUNAME, PROCTYPE, PSBNAME, PST, REGION, STARTTIME, SYSID, TRANID, and USERID

The following table provides variable information for the IMS system with a variable class of transaction:

Record Type	Variable Name	SMF Subtype	Description
PERFormance	LIFEtime	32, 34	Average transaction lifetime, in seconds
PERFormance	USE|COUNT	32, 34	Number of transactions executed
PERFormance	INQueuetime	32, 34	Average time, in seconds, spent in the input queue
PERFormance	OUTQueuetime	32, 34	Average time, in seconds, spent in the output queue
PERFormance	PROCtime	32, 34	Average time, in seconds, spent processing the transaction

IMS REGION Variable Class

This section describes variable information for class REGION on system IMS.

Syntax:

`[IMS] [record-type] REGION variable-name`

ID Options

The following ID options can be used with any IMS REGION variable: DBNAME, ENDTIME, IMSID, IMSREL, JOBNAME, LUNAME, PSBNAME, PST, STARTTIME, SYSID, TRANID, and USERID

The following table provides variable information for the IMS system with a variable class of region:

Record Type	Variable Name	SMF Subtype	Description
SUMmary	SCHEDULERtime	33, 35	Average elapsed scheduler processing time
SUMmary	INTENTwaittime	33, 35	Average intent contention wait time

Record Type	Variable Name	SMF Subtype	Description
SUMmary	POOLSPCwairtime	33, 35	Average pool space wait time
SUMmary	DB-IOTIME	33, 35	Total database I/O time

IMS DLI Variable Class

This section describes variable information for class DLI on system IMS.

Syntax:

`[IMS] [record-type] DLI variable-name`

ID Options

The following ID options can be used with any IMS DLI variable: DBNAME, ENDTIME, IMSID, IMSREL, JOBNAME, LUNAME, PSBNAME, PST, STARTTIME, SYSID, TRANID, and USERID.

The following table provides variable information for the IMS system with a variable class of DLI:

Record Type	Variable Name	SMF Subtype	Description
SUMmary	VSAMREADs	33, 35	Number of DL/I VSAM reads
SUMmary	VSAMWRITEs	33, 35	Number of DL/I VSAM writes
SUMmary	OSAMREADs	33, 35	Number of DL/I OSAM reads
SUMmary	OSAMWRITEs	33, 35	Number of DL/I OSAM writes
SUMmary	TOTALIO	33, 35	Total VSAM+OSAM DL/I I/O operations
SUMmary	DB-READs	33, 35	Amount of database reads
SUMmary	DB-WRITEs	33, 35	Amount of database writes
SUMmary	DB-GU	33, 35	Total database unique get requests
SUMmary	DB-GN	33, 35	Total requests for database get next
SUMmary	DB-GNP	33, 35	Total requests for database get next within parent
SUMmary	DB-GHU	33, 35	Number of requests for database get hold unique
SUMmary	DB-GHN	33, 35	Number of requests for database get hold next

Record Type	Variable Name	SMF Subtype	Description
SUMmary	DB-GHNP	33, 35	Number of requests for database get hold next within parent
SUMmary	DB-INSERT	33, 35	Number of database insert requests
SUMmary	DB-DELETE	33, 35	Number of database delete requests
SUMmary	DB-REPLACE	33, 35	Number of requests for database replace
SUMmary	DB-CALLS	33, 35	Number of DL/I database requests
SUMmary	MSG-GU	33, 35	Number of requests for message get unique
SUMmary	MSG-GN	33, 35	Number requests for message get next
SUMmary	MSG-INSERT	33, 35	Number of message insert calls
SUMmary	MSG-PURGE	33, 35	Number of message purge calls

MQ Variables

This section describes the MQ variables and the variable names for each class defined.

JOBNAME is the only ID option necessary, as you simply receive the same value as you would with object-related variables. Their usage lets you use them with JOBNAME.

MQ OBJEct Variable Class

This section describes variable information for class OBJECT class on system MQ.

Syntax:

`[MQ] [record-type] OBJEct variable-name`

ID Options

Use any MQ OBJEct variables with the following ID options: ENDTIME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information for the MQ system with a variable class of object:

Record Type	Variable Name	SMF Subtype	Description
REQuest	BROWSE	48	Get count with browse

Record Type	Variable Name	SMF Subtype	Description
REQuest	CLOSE	48	Close count for object
REQuest	GET	48	Get count for queue
REQuest	INQUIRE	48	Inquire count for object
REQuest	OPEN	48	Open count for object
REQuest	PUT	48	Put count for queue
REQuest	PUT1	48	Put1 count for queue
REQuest	SET	48	Set count for object

MQ OBJFailure Variable Class

This section describes variable information for class OBJFailure class on system MQ.

Syntax:

`[MQ] [`*`record-type`*`] OBJFailure `*`variable-name`*

ID Options

Use any MQ OBJFailure variables with the following ID options: ENDTIME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information:

Record Type	Variable Name	SMF Subtype	Description
REQuest	CLOSE	48	Close counts that failed
REQuest	GET	48	Get counts that failed
REQuest	INQUIRE	48	Inquire counts that failed
REQuest	OPEN	48	Open counts that failed
REQuest	PUT	48	Put counts that failed
REQuest	PUT1	48	Put1 counts that failed
REQuest	SET	48	Set counts that failed

MQ OBJMessage Variable Class

This section describes variable information for class OBJMessage class on system MQ.

Syntax:

`[MQ] [record-type] OBJMessage variable-name`

ID Options

Use any OBJMessage variables with the following ID options: ENDTIME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information:

Record Type	Variable Name	SMF Subtype	Description
REQuest	GETRES	48	Gets with the date and times of the Put.
REQuest	MAXGET	48	Maximum message size for the gets.
REQuest	MAXPUT	48	Maximum message size for the puts.
REQuest	MINGET	48	Minimum message size for the gets.
REQuest	MINPUT	48	Maximum message size for the puts.

MQ OBJTimer Variable Class

This section describes variable information for class OBJTimer class on system MQ.

Syntax:

`[MQ] [record-type] OBJTimer variable-name`

ID Options

Use any MQ OBJTimer variables with the following ID options: ENDTIME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information:

Record Type	Variable Name	SMF Subtype	Description
REQuest	GMSGTOT	48	Total message size for gets
REQuest	MAXCPU	48	Maximum CPU time for all requests
REQuest	MAXElapsed	48	Maximum elapsed time for all requests
REQuest	MAXRESID	48	Maximum residency time for all requests
REQuest	PMSGTOT	48	Total message size for puts

Record Type	Variable Name	SMF Subtype	Description
REQuest	TOTCPU	48	Total CPU time for all requests
REQuest	TOTElapsed	48	Total elapsed time for all requests
REQuest	TOTRESID	48	Total residency time for all requests

MQ JOBStat Variable Class

This section describes variable information for class JOBStat class on system MQ.

Syntax:

`[MQ] [record-type] JOBStat variable-name`

ID Options

Use any MQ JOBStat variables with the following ID options: ENDTIME, JOBNAME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information:

Record Type	Variable Name	SMF Subtype	Description
REQuest	BROWSE	48	Get count with browse
REQuest	CLOSE	48	Close count for job
REQuest	GET	48	Get count for job
REQuest	INQUIRE	48	Inquire count for job
REQuest	OPEN	48	Open count for job
REQuest	PUT	48	Put count for job
	PUT1		Put1 count for job
REQuest	SET	48	Set count for job

MQ JOBFailure Variable Class

This section describes variable information for class JOBFailure class on system MQ.

Syntax:

`[MQ] [record-type] JOBFailure variable-name`

ID Options

Use any MQ JOBFailure variables with the following ID options: ENDTIME, JOBNAME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information:

Record Type	Variable Name	SMF Subtype	Description
REQuest	CLOSE	48	Close counts that failed
REQuest	GET	48	Get counts that failed
REQuest	INQUIRE	48	Inquire counts that failed
REQuest	OPEN	48	Open counts that failed
REQuest	PUT	48	Put counts that failed
	PUT1		Put1 counts that failed
REQuest	SET	48	Set counts that failed

MQ JOBMessage Variable Class

This section describes variable information for class JOBMessage class on system MQ.

Syntax:

`[MQ] [`*`record-type`*`] JOBMessage `*`variable-name`*

ID Options

Use any JOBMessage variables with the following ID options: ENDTIME, JOBNAME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information:

Record Type	Variable Name	SMF Subtype	Description
REQuest	GETRES	48	Gets with the date and times of the Put
REQuest	MAXGET	48	Maximum message size for gets
REQuest	MAXPUT	48	Maximum message size for puts
REQuest	MINGET	48	Minimum message size for gets
REQuest	MINPUT	48	Maximum message size for puts

MQ JOBTimer Variable Class

This section describes variable information for class JOBTimer class on system MQ.

Syntax:

`[MQ] [record-type] JOBTimer variable-name`

ID Options

Use any MQ JOBTimer variables with the following ID options: ENDTIME, JOBNAME, OBJNAME, OBJSUBTYPE, OBJTYPE, QMANAGER, RESNAME and RESMANAGER, STARTTIME and SYSID.

The following table provides variable information:

Record Type	Variable Name	SMF Subtype	Description
REQuest	GMSGTOT	48	Total message size for gets
REQuest	MAXCPU	48	Maximum CPU time for all requests
REQuest	MAXElapsed	48	Maximum elapsed time for all requests
REQuest	MAXRESID	48	Maximum residency time for all requests
REQuest	PMSGTOT	48	Total message size for puts
REQuest	TOTCPU	48	Total CPU time for all requests
REQuest	TOTElapsed	48	Total elapsed time for all requests
REQuest	TOTRESID	48	Total residency time for all requests

AUDIT Variables

This section describes the AUDIT variables and the variable names for each class defined.

Note: Only one AUDIT variable is defined, and it has an equivalent in the ID options of AUDTYPE. The best way to write Audit reports is to use the FLASHBACK AUDITLOG command, or use the FLASHBACK LIST with two or more ID options.

AUDIT ALL Variable Class

This section describes variable information for the ALL class on the AUDIT system.

Syntax:

`[AUDIT][record-type] ALL variable name`

ID Options

Use any AUDIT ALL variables with the following ID options: STARTTIME, ENDTIME, AUDTYPE, SYSTEM, SUBSYS, AJOBID, TASKNAME, SECGROUP, RELEASE, AJOBNAME, AUSERID, ATERMID

Note: STARTTIME and ENDTIME in Audit reports represent the exception time.

The following table provides variable information for the AUDIT system with a variable class of ALL:

Record Type	Variable Name	SMF Subtype	Description
LOG	TYPE	01	Audit record type

COMMON Variables

This section describes the COMMON variables and the variable names for each class defined.

COMMON STATE Variable Class

This section describes variable information for class EXCEPTION class on the COMMON system.

Syntax:

`COMmon [record-type] EXCeption variable name`

ID Options

Use any COMmon STAte EXCeption variables with the following ID options: APPLID, CJOBID, GROUP, JOBNAME, OLDSTATUS, OWNER, RESRC, RESRC2, STARTTIME, STATUS, SUBGROUP, SYSTEM, VARNAME.

Use any COMmon THReshold EXCeption variables with the following ID options: APPLID, CJOBID, GROUP, JOBNAME, OLDSTATUS, OWNER, RESRC, RESRC2, RULETYPE, STARTTIME, STATUS, SUBGROUP, SYSTEM, THRECLAS, THRETYPE, TRANID, VARNAME.

Note: STARTTIME in common exception reports represents time of the exception.

Variable information is provided in the following table:

Record Type	Variable Name	SMF Subtype	Description
THRESHOLD, STATE	VARname	03, 04	Variable name
THRESHOLD, STATE	RESOURCE or RESRC	03, 04	Resource involved with the exception
THRESHOLD, STATE	RESOURCE2 or RESRC2	03, 04	Resource2 (eight characters)
THRESHOLD, STATE	RESOURCE2L or RESRC2L	03, 04	Resource2 Long (48 characters)
THRESHOLD, STATE	GROUP	03, 04	Group
THRESHOLD, STATE	SUBGroup	03, 04	Subgroup
THRESHOLD, STATE	VALUE	03, 04	Average threshold value that exceeded the limit or current state value
THRESHOLD	MAXVALUE	03	Maximum threshold value that exceeded the limit
THRESHOLD	MINVALUE	03	Minimum threshold value that exceeded the limit
THRESHOLD	WARNING	03	Warning limit value
THRESHOLD	PROBLEM	03	Problem limit value
THRESHOLD, STATE	STATUS	03, 04	Exception status
THRESHOLD, STATE	OLDVALUE	03, 04	Previous value
THRESHOLD	OLDWARNING	03	Previous warning limit value
THRESHOLD	OLDPROBLEM	03	Previous problem limit value

Record Type	Variable Name	SMF Subtype	Description
THRESHOLD, STATE	OLDSTATUS	03, 04	Previous status
THRESHOLD, STATE	ELAPSED	03, 04	Time elapsed from previous exception, in seconds
THRESHOLD, STATE	COUNT	03, 04	Exception count
THRESHOLD	RTYPE	03	Threshold rule type: None, Upper, Lower, Change
THRESHOLD	VTYPE	03	Variable type
THRESHOLD	TTYPE	03	Threshold type
THRESHOLD	CLASS	03	Count, Count K (count in 1024), Rate, Time, and Percent threshold

MVS Variables

The following pages describe the MVS variables and the variable names for each class defined for the z/OS operating system.

MVS ASCB Variable Class

This section describes variable information for class ASCB on system MVS.

Syntax:

`MVS [record-type] ASCB variable-name | data`

No ID options can be used with MVS ASCB variables.

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	AVG-BATCH	SMF70BTT	Average number of batch address spaces
SYSTEM	AVG-IN	SMF70ITT	Average number of address spaces not in storage and ready to execute
SYSTEM	AVG-OUT	SMF700TT	Average number of address spaces not in storage and ready to execute
SYSTEM	AVG-READY	SMF70RTT	Average number of address spaces in storage and ready to execute

Record Type	Variable Name	Data	Description
SYSTEM	AVG-STC	SMF70STT	Average number of started-task address spaces
SYSTEM	AVG-TSO	SMF70TTT	Average number of TSO address spaces
SYSTEM	AVG-WAIT	SMF70WTT	Average number of address spaces not in storage and not ready to execute
SYSTEM	MAX-BATCH	SMF70BMM	Maximum number of batch address spaces
SYSTEM	MAX-IN	SMF70IMM	Maximum number of address spaces in storage
SYSTEM	MAX-OUT	SMF70OMM	Maximum number of address spaces not in storage and ready to execute
SYSTEM	MAX-READY	SMF70RMM	Maximum number of address spaces in storage and ready to execute
SYSTEM	MAX-STC	SMF70SMM	Maximum number of started-task address spaces
SYSTEM	MAX-TSO	SMF70TMM	Maximum number of TSO address spaces
SYSTEM	MAX-WAIT	SMF70WMM	Maximum number of address spaces not in storage and not ready to execute
SYSTEM	MIN-BATCH	SMF70BMN	Minimum number of batch address spaces
SYSTEM	MIN-IN	SMF70IMN	Minimum number of address spaces in storage
SYSTEM	MIN-OUT	SMF700MN	Minimum number of address spaces not in storage and ready to execute
SYSTEM	MIN-READY	SMF70RMN	Minimum number of address spaces in storage and ready to execute
SYSTEM	MIN-STC	SMF70SMN	Minimum number of started-task address spaces
SYSTEM	MIN-TSO	SMF70TMN	Minimum number of TSO address spaces
SYSTEM	MIN-WAIT	SMF70WMN	Minimum number of address spaces not in storage and not ready to execute

MVS CHANNEL Variable Class

This section describes variable information for class CHANNEL on system MVS.

Syntax:

```
MVS [record-type] CHANNEL variable-name | data
```

ID Option

The following ID option can be used with the MVS CHANNEL variable: CHNID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYStem	LRead	SMF73PRU	LPAR rate of data read through the channel in bytes per second
SYStem	LWrite	SMF73PWU	LPAR rate of data written through the channel in bytes per second
SYStem	READ	SMF73TRU	Total rate of data read through the channel in bytes per second
SYStem	TIME	SMF73PTI	Measurement interval length in seconds
SYStem	WRITE	SMF73TWU	Total rate of data written through the channel in bytes per second
SYStem	%USE	SMF73TUC, SMF73TUT	Average percentage of time the channel was active

Note: To avoid rounding values too much, use the XTND variable option with rate variables of this class.

MVS CPU Variable Class

This section describes variable information for class CPU on system MVS.

Syntax:

`MVS [record-type] CPU variable-name | data`

ID Option

The following ID options can be used with MVS CPU variables: CPUID, CPUPID, and LPARNAME.

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	%BUSY|ACTIVE|UTILization	SMF70WAT	Percentage of time CPU was busy or active. If you are running PR/SM and this field displays as 0 percent or 100 percent, use the MVS LPAR variable instead.
SYSTEM	%WAIT	SMF70WAT	Percentage of time CPU was inactive
SYSTEM	BUSY|ACTIVE	SMF70WAT	Total time CPU was busy or active

Record Type	Variable Name	Data	Description
SYSTEM	WAIT	SMF70WAT	Total time CPU was inactive

MVS DEVICE Variable Class

This section describes variable information for class DEVICE on system MVS.

Syntax:

`MVS [record-type] DEVICE variable-name | data`

ID Option

The following ID option can be used with MVS DEVICE variables: DEVID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	CONNECT	SMF74CNN	Average time spent connected to a channel path
SYSTEM	CUB-DELAY	SMF74CUB	Average time I/O request is delayed due to busy control unit
SYSTEM	DB-DELAY	SMF74DVB	Average time I/O request is delayed due to busy device
SYSTEM	DISCONNECT	SMF74DIS	Average time spent disconnected during an I/O request
SYSTEM	PENDING	SMF74PEN	Average time spent waiting on hardware
SYSTEM	SERVTIME	SMF74ATV	Average service time for device to complete an I/O request
SYSTEM	SSCH	SMF74SSC	Number of start subchannels
SYSTEM	UTILization	SMF74ATV	Percentage of time device was busy or active

MVS DISK Variable Class

This section describes variable information for class DISK on system MVS.

Syntax:

`MVS [record-type] DISK variable-name | data`

ID Option

The following ID options can be used with MVS DISK variables: DEVID, DISKID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	CONNECT	SMF74CNN	Average time spent connected to a channel path
SYSTEM	CUB-DELAY	SMF74CUB	Average time I/O request is delayed due to busy control unit
SYSTEM	DB-DELAY	SMF74DVB	Average time I/O request is delayed due to busy device
SYSTEM	DISCONNECT	SMF74DIS	Average time spent disconnected during an I/O request
SYSTEM	PENDING	SMF74PEN	Average time spent waiting on hardware
SYSTEM	SERVTIME	SMF74ATV	Average service time for device to complete an I/O request
SYSTEM	SSCH	SMF74SSC	Number of start subchannels
SYSTEM	UTILization	SMF74ATV	Percentage of time disk was busy or active

MVS DSN Variable Class

This section describes variable information for class DSN on system MVS.

Syntax:

`MVS [record-type] DSN variable-name | data`

ID Option

The following ID options can be used with MVS DSN variables:

DEVID, DISKID, DSNID, JOBID, STEPID, VOLSER

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	ACTIVITY	SMF64	Number of VSAM requests
SYSTEM	BPR	SMF64	Average buffer performance rating for VSAM data sets
SYSTEM	CA-SPLITS	SMF64DAS	Number of control area splits for VSAM data sets
SYSTEM	CI-SPLITS	SMF64DCS	Number of control interval splits for VSAM data sets
SYSTEM	DELETES	SMF64DDE	Number of VSAM deletes
SYSTEM	EXCP	SMF64, SMF14, SMF15	Total number of EXCPs

Record Type	Variable Name	Data	Description
SYSTEM	EXCP-VSAM	SMF64DEP	Number of EXCPs for VSAM data sets
SYSTEM	EXCP-NVSM	SMF14, SMF15, SMFEXCP	Number of EXCPs for non-VSAM data sets
SYSTEM	INPUT	SMF14, SMFEXCP	Number of EXCPs for non-VSAM data sets
SYSTEM	INSERTS	SMF64DIN	Number of VSAM inserts
SYSTEM	OUTPUT	SMF15, SMFEXCP	Number of EXCPs for non-VSAM output data set
SYSTEM	RETRIEVES	SMF64DRE	Number of VSAM retrieves
SYSTEM	UPDATES	SMF64DUP	Number of VSAM updates
SYSTEM	VOLSER	SMF64, SMF14, SMF15	Volume serial number of device.

MVS EXCP Variable Class

This section describes variable information for class EXCP on system MVS.

Syntax:

`MVS [`*`record-type`*`] EXCP `*`variable-name`*` | `*`data`*

ID Option

The record type of an EXCP variable determines which ID options can be used with the variable, as follows:

- The following ID options can be used with MVS JOB EXCP variables:

  ACCTID, DDNAME, DEVID, JOBCLASS, JOBID, PGID, ROGRAMMER, USERID

- The following ID options can be used with MVS STEP EXCP variables:

  ACCTID, DDNAME, DEVID, JOBCLASS, JOBID, PGID, PROGID, PROGRAMMER, STEPID, USERID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
JOB, STEP	COUNT	SMF30BLK	Number of EXCPs issued
JOB, STEP	DDNAME	SMF30DDN	DDname
JOB, STEP	DEVCUU	SMF30CUA	Device number

Record Type	Variable Name	Data	Description
JOB, STEP	DEVTYPE	SMF30DEV	Device class
JOB, STEP	DEV-CONN	SMF30DCT	Total device connect time
JOB, STEP	LBLKSIZE	SMF30BSZ	Largest block size
JOB, STEP	UNITTYPE	SMF30UTP	Device unit type

MVS IO Variable Class

This section describes variable information for class IO on system MVS.

Syntax:

`MVS [record-type] IO variable-name | data`

ID Option

The record type of an IO variable determines which ID options can be used with the variable, as follows:

- No ID options can be used with MVS SYSTEM IO variables.
- The following ID options can be used with MVS JOB IO and MVS STEP IO variables: JOBID, STEPID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
JOB, STEP	CARDs	SMF30INP	Number of card-image records read
JOB, STEP	DEVCON	SMF30TCN	Total device connect time
JOB, STEP	EXCPs	SMF30TEP	Number of EXCPs
SYSTEM	INTERRUPTs	SMF70	Total number of I/O interrupts
SYSTEM	INT-SLIH	SMF70SLH	Number of I/O interrupts handled by the I/O interrupt handler
SYSTEM	INT-TPI	SMF70TPI	Number of I/O interrupts handled through the TPI instruction
JOB, STEP	TGETs	SMF30TGT	Number of TGETs
JOB, STEP	TPUTs	SMF30TPT	Number of TPUTs

MVS JOB and STEP Variable Classes

This section describes variable information for classes JOB and STEP on system MVS.

Syntax:

MVS [*record-type*] *variable-class variable-name* | *data*

ID Option

- The following ID options can be used with z/OS JOB variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGRAMMER, USERID
- The following ID options can be used with z/OS STEP variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGID, PROGRAMMER, STEPID, USERID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	%CPUACT	SMF30	Percentage of CPU consumed while active
SYSTEM	%CPUDUR	SMF30	Percentage of CPU consumed during the JOB or STEP
SYSTEM	%CPUINT	SMF30	Percentage of CPU consumed in report interval
SYSTEM	%CPURES	SMF30	Percentage of CPU consumed while resident
SYSTEM	ACTIVE	SMF30	Total transaction active time
SYSTEM	COUNT	SMF30	Number of jobs or steps executed
SYSTEM	CPU	SMF30	Total amount of CPU time consumed
SYSTEM	DEV-CONN	SMF30	Total device connect time
SYSTEM	DURATION	SMF30	Total job or step duration time
SYSTEM	RESIDENT	SMF30	Total transaction residency time
SYSTEM	SAMPLEs	SMF30	Number of step interval samples
SYSTEM	SRB	SMF30	Total amount of SRB CPU time consumed
SYSTEM	TCB	SMF30	Total amount of TCB CPU time consumed
SYSTEM	TOT-EXCP	SMF30	Number of EXCPs

MVS LPAR Variable Class

This section describes variable information for class LPAR on system MVS.

Syntax:

`MVS [record-type] LPAR variable-name | data`

ID Option

The following ID options can be used with MVS LPAR variables:

CPUID, CPUPID, LPARID, LPID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	%ACTIVE	SMF70PDT	Logical processor usage
SYSTEM	ACTIVE	SMF70PDT	Logical processor active time
SYSTEM	CPU	SMF70VPA	Logical processor address
SYSTEM	NUMBER	SMF70LPN	Logical partition number
SYSTEM	RWF	SMF70BPS	Logical partition resource weight factor

MVS OPERator Variable Class

This section describes variable information for class OPERATOR on system MVS.

Syntax:

`MVS [record-type] OPER variable-name | data`

ID Option

The record type of an OPERator variable determines which ID options can be used with the variable, as follows:

- The following ID options can be used with MVS JOB OPERator variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGRAMMER, USERID

- The following ID options can be used with MVS STEP OPERator variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGID, PROGRAMMER, STEPID, USERID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
JOB, STEP	NS-DASD-MOUNTs	SMF30PDM	Number of nonspecific DASD mounts
JOB, STEP	NS-MSS-MOUNTs	SMF30MTM	Number of nonspecific MTM mounts
JOB, STEP	NS-TAPE-MOUNTs	SMF30PTM	Number of nonspecific tape mounts
JOB, STEP	S-DASD-MOUNTs	SMF30PRD	Number of specific DASD mounts
JOB, STEP	S-MSS-MOUNTs	SMF30MSR	Number of specific MTM mounts
JOB, STEP	S-TAPE-MOUNTs	SMF30TPR	Number of specific tape mounts

MVS PAGE Variable Class

This section describes variable information for class PAGE on system MVS.

Syntax:

```
MVS [record-type] PAGE variable-name | data
```

ID Option

The record type of a PAGE variable determines which ID options can be used with the variable, as follows:

- No ID options can be used with MVS SYSTEM PAGE variables.
- The following ID options can be used with MVS JOB PAGE variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGRAMMER, USER
- The following ID options can be used with MVS STEP PAGE variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGID, PROGRAMMER, STEPID, USERID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	AVG-UIC	SMF71ACA	Average of the maximum unreferenced interval count during any RMF interval
SYSTEM, JOB, STEP	IN-COMMON	SMF71SNI, SMF30CPI	Number of common area page-ins
SYSTEM, JOB, STEP	IN-LPA	SMF71LNI, SMF30IPI	Number of LPA page-ins

Record Type	Variable Name	Data	Description
SYSTEM, JOB, STEP	IN-PRIvate	SMF71PIN, SMF30PGI	Number of private area page-ins
SYSTEM, JOB, STEP	IN-SWAP	SMF71SIN, SMF30PSI	Number of pages swapped in
SYSTEM, JOB, STEP	IN-VIO	SMF71VIN, SMF30VPI	Number of VIO page-ins
SYSTEM, JOB, STEP	INs	SMF71, SMF30	Total number of page-in operations
JOB, STEP	IOs	SMF30	Total number of page operations
SYSTEM	MAX-UIC	SMF71LIC	Lowest maximum unreferenced interval count during any RMF interval
SYSTEM	MIGRATE	SMF71PEA	Number of pages migrated from expanded storage
SYSTEM	MIN-UIC	SMF71LIC	Lowest maximum unreferenced interval count during any RMF interval
SYSTEM	MOVES	SMF71PMV	Number of pages moved between storage above the 16M line and storage below the 16M line
SYSTEM	MOVES-EXT	SMF71PES	Number of pages moved to expanded storage
SYSTEM, JOB, STEP	OUT-COMMON	SMF71SNO, SMF30CPR	Number of page-outs performed in the common area
SYSTEM, JOB, STEP	OUT-PRIvate	SMF71POT, SMF30PGO	Number of private area page-outs
SYSTEM, JOB, STEP	OUT-SWAP	SMF71SOT, SMF30PSO	Number of pages swapped out
SYSTEM, JOB, STEP	OUT-VIO	SMF71VOT, SMF30VPO	Number of VIO page-outs
SYSTEM, JOB, STEP	OUTs	SMF71, SMF30	Total number of page-out operations
SYSTEM, JOB, STEP	RECLAIM-COMMON	SMF71SNR, SMF30CPR	Number of common areas reclaimed
SYSTEM, JOB, STEP	RECLAIM-LPA	SMF71LNR, SMF30LPR	Number of LPA reclaims
SYSTEM, JOB, STEP	RECLAIM-PRIvate	SMF71PRC, SMF30REC	Number of private areas reclaimed
SYSTEM, JOB, STEP	RECLAIM-VIO	SMF71VRC, SMF30VPR	Number of VIO reclaims

Record Type	Variable Name	Data	Description
SYSTEM, JOB, STEP	RECLAIMs	SMF71, SMF30	Total number of pages reclaimed
JOB, STEP	SECONDs	SMF30PSC	Number of CPU page seconds
JOB, STEP	STEALs	SMF30PST	Number of pages stolen

MVS PAGESWAP Variable Class

This section describes variable information for class PAGESWAP on system MVS.

Syntax:

```
MVS [record-type] PAGESWAP variable-name | data
```

ID Option

The following ID option can be used with MVS PAGESWAP variable: DSNID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	AVGUSED	SMF75AVU	Maximum number of slots or swap-sets used
SYSTEM	IO	SMF75SIO	Number of I/O requests
SYSTEM	MAXUSED	SMF75MXU	Maximum number of slots or swap-sets used
SYSTEM	MINUSED	SMF75MNU	Minimum number of slots or swap-sets used
SYSTEM	PAGES	SMF75PGX	Number of pages transferred to or from page data set
SYSTEM	SLOTS	SMF75SLA	Number of slots or swap-sets within the page or swap data set
SYSTEM	UNUSABLE	SMF75BDS	Number of unusable slots or swap-sets

MVS PERFormance Variable Class

This section describes variable information for class PERFORMANCE on system MVS.

Syntax:

```
MVS [record-type] PERF variable-name | data
```

ID Option

The record type of a PERFormance variable determines which ID options can be used with the variable, as follows:

- The following ID options can be used with MVS JOB PERFormance variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGRAMMER, USERID

- The following ID options can be used with MVS STEP PERFormance variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGID, PROGRAMMER, STEPID, USERID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
JOB, STEP	ACTIVE	SMF30TAT	Total transaction active time
JOB, STEP	CPU-SU	SMF30CSU	Number of CPU service units
JOB, STEP	IOC-SU	SMF300	Number of I/O service units
JOB, STEP	MSO-SU	SMF30MSO	Number of main storage service units
JOB, STEP	RESIDENT	SMF30RES	Total transaction residency time
JOB, STEP	SRB-SU	SMF30SRB	Number of SRB service units
JOB, STEP	TOT-SU	SMF30RSV	Total number of service units

MVS PRINT Variable Class

This section describes variable information for class PRINT on system MVS.

Syntax:

`MVS [`*`record-type`*`] PRINT `*`variable-name`*` | `*`data`*

ID Option

The following ID options can be used with MVS PRINT variables:

SMF6FMN, SMF6OWC

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	SMF6FMN	SMF6FMN	Form number
SYSTEM	SMF6NLR	SMF6NLR	Number of logical records written

SYSTEM	SMF6OUT	SMF6OUT	Output device name
SYSTEM	SMF6OWC	SMF6OWC	SYSOUT class

MVS PROCessor Variable Class

This section describes variable information for class PROCESSOR on system MVS.

Syntax:

```
MVS [record-type] PROC variable-name | data
```

ID Option

The record type of a PROCessor variable determines which ID options can be used with the variable, as follows:

- The following ID options can be used with MVS JOB PROCessor variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGRAMMER, USERID

- The following ID options can be used with MVS STEP PROCessor variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGID, PROGRAMMER, STEPID, USERID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
JOB, STEP	CONTROL-TASK-CPU (ESA)	SMF30RCT	Total transaction active time
JOB, STEP	HIPERSPACE (ESA)	SMF30HPT	Hiperspace processing time
JOB, STEP	INITIATOR-SRB	SMF30ISB	Initiator CPU time under SRB
JOB, STEP	INITIATOR-TCB	SMF30ICU	Initiator CPU time under TCB
JOB, STEP	INITIATOR-VECTOR-AFFINITY	SMF30IVA	Total transaction residency time
JOB, STEP	INITIATOR-VECTOR-USAGE	SMF30IVU	Initiator vector usage time
JOB, STEP	IO-INT-CPU (ESA)	SMF30IIP	CPU time used to process I/O interrupts
JOB, STEP	SRB	SMF30CPS	CPU time under SRB
JOB, STEP	TCB	SMF30CPT	CPU time under TCB
JOB, STEP	VECTOR-AFFINITY	SMF30FVA	Vector affinity time
JOB, STEP	VECTOR-USAGE	SMF30JVU	Vector usage time

MVS STORage Variable Class

This section describes variable information for class STORAGE on system MVS.

Syntax:

`MVS [record-type] STOR variable-name | data`

ID Option

The record type of a STORage variable determines which ID options can be used with the variable, as follows:

- The following ID options can be used with MVS JOB STORage variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGRAMMER, USERID

- The following ID options can be used with MVS STEP STORage variables:

  ACCTID, JOBCLASS, JOBID, PGID, PROGID, PROGRAMMER, STEPID, USERID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
JOB, STEP	ABOVE	SMF30EUR	Virtual storage allocated from user subpools above the 16-MB line
JOB, STEP	BELOW	SMF30URB	Virtual storage allocated from user subpools below the 16-MB line
JOB, STEP	LSQA/SWA-ABOVE	SMF30EAR	Virtual storage allocated from LSQA/SWA subpools above the 16-MB line
JOB, STEP	LSQA/SWA-BELOW	SMF30ARB	Virtual storage allocated from LSQA/SWA subpools below the 16-MB line
JOB, STEP	PRIVATE-ABOVE	SMF30ERG	Private area size above the 16-MB line
JOB, STEP	PRIVATE-BELOW	SMF30RGB	Private area size below the 16-MB line
JOB, STEP	PRIVATE-HIGH	SMF30SYS	Amount of storage used from top of private area
JOB, STEP	PRIVATE-LOW	SMF30PRV	Amount of storage used from bottom of private area
JOB, STEP	REGION-SIZE	SMF30ORGN	Region size

MVS SU Variable Class

This section describes variable information for class SU (service units) on system MVS.

Syntax:

`MVS [record-type] SU variable-name | data`

ID Option

The following ID options can be used with MVS SU variables: PGDID, PGID,

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	CPU	SMF72CTS	Number of CPU service units consumed
SYSTEM	IOC	SMF72ITS	Number of I/O service units consumed
SYSTEM	MSO	SMF72MTS	Number of storage service units consumed
SYSTEM	SRB	SMF72STS	Number of SRB service units consumed
SYSTEM	TOTAL	SMF72	Total number of service units consumed

MVS SWAP Variable Class

This section describes variable information for class SWAP on system MVS.

Syntax:

`MVS [record-type] SWAP variable-name | data`

ID Option

The following ID option can be used with MVS SWAP variables: SWAPID

Variable information is provided in the following table:

Record Type	Variable Name	Data	Description
SYSTEM	COUNT	SMF71CNT	Total number of swap operations
SYSTEM	L-AUXiliary	SMF71LAX	Number of logical swaps physically swapped to auxiliary storage
SYSTEM	L-EXTended	SMF71LES	Number of logical swaps physically swapped to extended storage

Record Type	Variable Name	Data	Description
SYSTEM	MIGRATEs	SMF71MIG	Number of physical swaps migrated from extended storage to auxiliary storage
SYSTEM	P-AUXiliary	SMF71AXD	Number of physical swaps directed to auxiliary storage
SYSTEM	P-EXTended	SMF71ESD	Number of physical swaps directed to extended storage

MVS TSO Variable Class

This section describes variable information for class TSO on system MVS. These variables are applicable only to SMF32, subtype 3.

Syntax:

`MVS [record-type] TSO variable-name | data`

ID Options

The following ID options can be used with MVS TSO variables: JOBID, STEPID, TSOCMD

Variable information is provided in the following table.

Record Type	Variable Name	Data	Description
SYSTEM	DEV-CON	SMF32TCT	Total device connect time by TSO user
SYSTEM	EXCP	SMF32EXP	Number of EXCPs issued by TSO user
SYSTEM	SRB	SMF32SRB	SRB time consumed by TSO user
SYSTEM	TCB	SMF32TCB	TCB time consumed by TSO user
SYSTEM	TGET	SMF32TGT	Number of TGETs by TSO user
SYSTEM	TPUT	SMF32TPT	Number of TPUTs by TSO user
SYSTEM	TRAN-CNT	SMF32CNT	Number of transactions by TSO user

Chapter 6: Variable Options

This section contains the following topics:

Description and Syntax (see page 157)
COUNT Option (see page 158)
FORMAT Option (see page 158)
LOGICAL Option (see page 159)
MAX and MIN Options (see page 159)
MAXDATE and MAXTIME Options (see page 160)
PerCenT Option (see page 161)
RATE Option (see page 161)
SELZERO Option (see page 163)
TOTAL Option (see page 163)
XTIME Option (see page 164)
XTND and NXTND Options (see page 165)

Description and Syntax

Two kinds of options can be used with variables: variable options and ID options. This section describes the variable options that can be used with most variables.

The following variable options are valid only in tabular reports:

- DIFF
- RANGE
- RIGHT
- SELECT

Specify variable options following the variable to which they are to apply, as the following syntax shows:

variable [*option* [*option*...]]

More information:

ID Options (see page 167)
Tabular Reports (see page 191)

COUNT Option

Use the COUNT option to convert a variable into the number of different identifiers that correspond to that variable.

The COUNT option has the following syntax:

```
COUNT=ID option
```

ID*option*

Includes any ID option valid for the variable.

Example: COUNT Option

The following command would produce a tabular report displaying the number of terminals that had any activity:

```
TAB TRAN USE COUNT=TERMID
END
RUN
```

FORMAT Option

Use the FORMAT option to define the length of an output field or to specify how many decimal positions it contains.

The FORMAT option can also be used with ID options to control the number of characters printed for each identifier. For more information about formatting identifiers, see the chapter “ID Options.”

The FORMAT option has the following syntax:

```
FORMAT (outputformat)
```

outputformat

Specifies one of the following, for numeric or alphanumeric output:

For numeric data: Specifies up to 15 *N*s with or without a period to indicate the placement of the decimal point. For example, FORMAT(NN.NNN) indicates a five-digit output rounded to three decimal places. You can include any constants other than the letters *N* and *C* in the string. For example, specify FORMAT(NN.NNN%) to include a percent sign with each value in your report.

For alphanumeric data: Specifies *n*C, where n is the number of characters to appear in the output. For example, FORMAT(4C) indicates an output of four characters. The value of *n has* no limit.

Note: If a value for a variable is greater than the output format you specify, an asterisk replaces the value. For example, if you specify FORMAT(*NN.NN*) with a variable, asterisks replace all values greater than 99.99 for that variable in your report.

Examples: FORMAT Option

The following are FORMAT option examples and the result of each:

TRAN LIFE FORMAT(*NNN.NN*)

By default, the output of the variable TRAN LIFE appears in the format *N.NNN, NN.NN,* or *NNNNN,* depending on the size of the value. The FORMAT option here forces the output to two decimal places.

TRAN USE RATE FORMAT(*NN.N*)

The RATE option used with the variable TRAN USE defaults to two decimal positions. The FORMAT option here alters the output to one decimal place.

LOGICAL Option

By default, the Report Writer reviews all input records and discards any with negative values. Use the LOGICAL option with a variable to bypass this check and treat all values for the variable as unsigned (logical) values.

The LOGICAL option has the following syntax:

```
LOGICAL
```

Example: LOGICAL Option

The following commands create a tabular report on terminals, treating all values for the variable TERM USE as unsigned. Records with a negative value for TERM RESP are still discarded.

```
TAB TERMID TERM RESP AND,
           TERM USE LOGICAL
END
RUN
```

MAX and MIN Options

The MAX and MIN options forces a variable that by default prints a count or an average to print the minimum or maximum value.

The MAX and MIN options have the following syntax:

```
MAX[=idoption]
MIN[=idoption]
```

idoption

Includes any ID option valid for the variable. The values produced by the MAX and MIN options are largely dependent on the structure of the data records. Use this operand to control how the values are computed.

Examples: MAX and MIN Options

The following examples use the MAX and MIN options:

```
TRAN LIFE MAX
TRAN LIFE MIN
DATASET USE MAX=TRANNUM
```

TRAN LIFE is defined as the average transaction lifetime. TRAN LIFE MAX prints the maximum transaction lifetime. TRAN LIFE MIN prints the minimum transaction lifetime.

DATASET USE is defined as the number of file requests. DATASET USE MAX=TRANNUM prints the maximum number of file requests for any single transaction number.

MAXDATE and MAXTIME Options

Use the MAXDATE and MAXTIME options to print the date and time when the value of a variable is highest.

The MAXDATE and MAXTIME options have the following syntax:

```
MAXDATE
MAXTIME
```

Examples: MAXDATE and MAXTIME Options

The following examples use the MAXDATE and MAXTIME options with the variable TRAN LIFE:

```
TRAN LIFE MAXDATE
TRAN LIFE MAXTIME
```

TRAN LIFE is defined as the average transaction lifetime. TRAN LIFE MAXDATE prints the date on which transaction lifetime was highest. TRAN LIFE MAXTIME prints the time at which this value was highest.

PerCenT Option

Use the PerCenT option to cause a variable to represent the usage of a resource as a percentage of the total value for the system. You can use the PerCenT option only with variables that represent a total that is *not* the same as the value for the entire system.

The PerCenT option has the following syntax:

```
PerCenT
```

Example: PerCenT Option

In the following example, the PLOT command plots the use of the transaction CINQ against the total system transaction use. The second command, PLOT2, plots the percentage of transaction use for CINQ compared to the total system transaction use.

```
PLOT  TRANSACTION USE     TRANID(CINQ)
PLOT2 TRANSACTION USE PCT TRANID(CINQ)
END
RUN
```

RATE Option

Use the RATE option to display a rate (number per second, minute, hour, day, or week) rather than a count. You can use the RATE option only with variables that represent a count.

The RATE option has the following syntax:

```
RATE[(period [, WEIGHTED] [type])]
```

period

Defines the time interval to base the rate upon. Specify one of the following:

(Default) RPTINT

Calculates the rate based on the reporting interval.

SYSINT

Calculates the rate based on the interval of the main system record, accounting only for the time CA SYSVIEW was active logging data.

[*nn*] *interval*

Calculates the rate based on the interval specified. The value of *nn* can be 1 through 99; the default is 1. Replace *interval* with SECONDs, MINUTEs, HOURs, DAYs, or WEEKs.

WEIGHTED

Prints rates based only on those intervals that show activity for the specified variable.

type

Specify one of the following:

COUNT-Prints the number of active intervals.

MAX-Prints the maximum value for any one interval.

MAXDATE-Prints the date of the interval during which the maximum value occurred.

MAXRATE-Prints the maximum rate for any one interval.

MAXTIME-Prints the time of the interval during which the maximum value occurred.

Examples: RATE Option

The PLOT command plots the use of the transaction CINQ against the total system transaction use. The PLOT2 command plots the transaction rate for CINQ compared to the total system transaction rate:

```
PLOT  TRAN USE      TRANID(CINQ)
PLOT2 TRAN USE RATE TRANID(CINQ)
  EACH HOUR
END
RUN
```

These commands create a tabular report showing the hourly transaction rate (average number per hour) each day:

```
TAB DATE TRAN USE RATE(1 HOUR)
  EACH DAY
END
RUN
```

These commands create a tabular report showing the highest hourly transaction rate for each day:

```
TAB DATE TRAN USE RATE(1 HOUR,MAXRATE)
  EACH DAY
END
RUN
```

SELZERO Option

By default, the Report Writer reviews all input records and discards any with zero activity for the specified variables. For example, a tabular report on terminal activity includes only terminals with a positive value for at least one variable specified with the TAB command. Use the SELZERO option to bypass this check and include all resources in the report.

The SELZERO option has the following syntax:

```
SELZERO
```

Example: SELZERO Option

By default, a graph of file requests by transaction ID includes only transactions that issued at least one file request. The following commands graph the average number of file requests by every transaction, including transactions with no file requests:

```
GRAF TRANID TRAN USE SELZERO
END
RUN
```

TOTAL Option

Use the TOTAL option to force a variable that by default prints an average to print a total.

The TOTAL option has the following syntax:

```
TOTAL
```

Example: TOTAL Option

The following example uses the TOTAL option with the variable TRAN CPU:

```
TRAN CPU TOTAL
```

By default, the variable TRAN CPU creates a column showing the average CPU time per transaction. Specifying the variable with the option TOTAL creates a column showing the total CPU time used by all transactions.

XTIME Option

By default, all time-based Report Writer variables are displayed in seconds. If you expect a variable to show a time of many hours, use the XTIME option to translate the time to *hhh:mm:ss* format.

- *hhh* = hours
- *mm* = minutes
- *ss* = seconds

If you want more precise output of time values, use one of the following options:

- XTIMEM option

  Translates the time value to *hhh:mm:ss:lll* format.

  lll = milliseconds.

- NXTIME option

  Changes the value to *ssss.lll* format, where *ssss* is seconds and *lll* is milliseconds.

Note: The *hhh:mm:ss, hhh:mm:ss:lll,* and *ssss.lll* formats take up more column space than the default format.

The XTIME option has the following syntax:

```
XTIME
XTIMEM
NXTIME
```

Example: XTIME Option

The following example uses the XTIME option with the variable TRAN TCPU:

```
TRAN TCPU XTIME
```

XTND and NXTND Options

Use the XTND and NXTND options to increase or decrease the number of digits in the output of a variable.

Most variables have a maximum value of five digits. If a number is larger than 99999, the Report Writer divides it by 1000 and adds a *K*. For example, 200,000 becomes 200 KB. If the number is still too large, the Report Writer divides by 1000 again and adds an *M*, and so on.

For values representing storage, the Report Writer divides large values by 1024 and adds a *k*. If the number is still too large, it divides by 1024 again and adds an *m.*

The XTND and NXTND variable options are *not* valid for variables that yield time values. You can use the XTIME, XTIMEM, and NXTIME options to control the output of time variables.

XTND Option

Use the XTND (extend) option to support the printing of up to eight digits. This ability allows you to print more precise output of numbers up to 99,999,999 (including commas).

The XTND option has the following syntax:

```
XTND
```

Example: XTND Option

The following XTND option with the variable TRAN USE gives an exact count when you expect the number of transactions to be more than 99999:

```
TRAN USE XTND
```

NXTND Option

Some numbers that the Report Writer expects to be large can be extended by default. Use the NXTND option to reduce the number of digits printed, thus increasing the number of columns that can fit across a page.

The NXTND option has the following syntax:

```
NXTND
```

Example: NXTND Option

The following example uses the NXTND option with the variable TRAN USE:

```
TRAN USE NXTND
```

Chapter 7: ID Options

This section contains the following topics:

How to Use ID Options (see page 167)
Descriptions of ID Options (see page 168)
ID Options Used with Commands (see page 177)
Use ID Options with Variables and Identifiers (see page 181)
ID Options Used with Commands or Variables (see page 182)
Generate a Plot Report Using Variables (see page 184)
Generate a Tab Report Using Commands and Variables (see page 185)
Identifiers (see page 186)
How Format Identifiers Work (see page 187)

How to Use ID Options

You can use ID options with certain commands, variables, or both, as follows:

- Use ID options with commands to organize the data in a tabular, graph, or flashback report by a specific type of resource. For example, you can create a tabular report organized by transaction ID in which each row contains information about only a single transaction.

  You can also specify one or more *identifiers* to limit the report to the particular resource or resources specified by the identifiers.

- Use ID options with variables in any type of report to limit the range of the variable to a particular resource or group of resources. For example, you can limit the range of a variable to a specific transaction or a specific group of programs.

  When you use ID options with variables, you generally must include one or more *identifiers* that specify the particular resources on which you want information.

More information:

Identifiers (see page 186)

Descriptions of ID Options

The tables in the following topics list:

- All ID options for CICS, COMMON, AUDIT, IMS, MQ, and MVS
- The identifiers that are valid for each
- Their maximum length in characters
- The variable record types they can be used with

The following list contains the record types and their SMF record subtypes:

- The LOG record type with AUDIT variables has an SMF record subtype of 01.
- The EXCEPTION record type with COMMON STATE variables has an SMF record subtype of 04.
- The SYSTEM record type with THRESHOLD variables has an SMF record subtype of 08.
- The INTERVAL record type has an SMF record subtype of 23 and 28.
- The SYSTEM record type with EXCEPTION variables has an SMF record subtype of 24.
- The SUMMARY record type with CICS variables has an SMF record subtype of 25.
- The PERFORMANCE record type with CICS variables has an SMF record subtype of 27.
- The PERFORMANCE record type with IMS variables has an SMF record subtype of 32.
- The SUMMARY record type with IMS variables has an SMF record subtype of 33.

ID Options for CICS

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
ABCODE	Abend codes for transactions	4	Any PERFormance variable
APPLID	VTAM APPLIDs of CICS regions of the CICS system	8	Only PERFormance variable
CICSREL	CICS release level in the format **v.r.m.** Where *v* is the version, *r* is the release, and *m* is the maintenance level.	5	Any INTerval, SYStem, and SUMmary variable
DB2PROG	DB2 requesting programs	8	PERFormance DB2
DB2TYPE	DB2 statement types	24	PERFormance DB2

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
DB2PSNUM	DB2 precompiler statement numbers	5	PERFormance DB2
DCTID	Data set names of transient data sets	4	PERFormance TERMinal, PERFormance TRANsaction, PERFormance TRANSIENT
ENDTIME	None	8 (*hh:mm:ss* format)	All variable classes
FILEID	File name	8	PERFormance DATASET and PERFormance DLI
JOBNAME	CICS jobname	8	Any CICS variable
NETNAME	VTAM logical unit names for terminals	8	Any PERFormance variable
OPERNAME	Operator names	20	Any PERFormance variable (release 4 only)
PROGID	Program names	8	PERFormance DATASET, PERFormance PROGram, PERFormance TERMinal, PERFormance TRANsaction
STARTTIME	None	8 (*hh:mm:ss* format)	All variable classes
SYSID	MVS system ID	4	All variable classes
TERMID	Terminal IDs	4	Any PERFormance and SYStem variable
THRCLAS	Threshold classes. Valid values are: COUNT, RATE, TIME, PCT	5	PERFormance TERMinal, PERFormance THRESHOLD, PERFormance TRANsaction
THRSID	Threshold names	8	PERFormance TERMinal, PERFormance THRESHOLD, PERFormance TRANsaction, SYStem THRESHOLD
THRSSID	Secondary threshold names	8	PERFormance TERMinal, PERFormance THRESHOLD, PERFormance TRANsaction
THRTYPE	Threshold types. Valid values are UPPER, LOWER	5	PERFormance TERMinal, PERFormance THRESHOLD, PERFormance TRANsaction
TRANID	Transaction IDs	4	Any PERFormance, SYStem, and SUMmary variable

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
TRANNUM	Transaction numbers	7 (exactly)	Any PERFormance variable and SYStem THRESHOLD
TRSCEID	Threshold resource names	8	PERFormance TERMinal, PERFormance THRESHOLD, PERFormance TRANsaction
TSTGID	Temporary storage IDs for temporary storage data	8	PERFormance TEMPSTOR, PERFormance TERMinal, PERFormance TRANsaction
USERID	User IDs	8	Any PERFormance and SYStem variable
UTRANID	Umbrella names	32	Any PERFormance variable
VOLSER	Volume serial number	8	PERFormance DATASET
VTAMLU	VTAM logical unit names of terminals	8	Any PERFormance variable and SYStem EXCEPTION

ID Options for IMS

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
DBNAME	Database name	8	Any SUMmary variable
ENDTIME	None	8 (*hh:mm:ss* format)	Any SUMmary variable
IMSID	IMS subsystem ID.	4	All variable classes
IMSREL	IMS release level in the format v.r, where v is the version, and r is the release	4	All variable classes
JOBNAME	IMS job names	8	All variable classes
LTERM	Logical terminal name	8	Any PERFormance variable
LUNAME	Logical unit name	8	All variable classes
PROCTYPE	Transaction processing type. Valid values are: A, B, C, D, O, P, Q, R, S, T, X	1	Any PERFormance variable
PSBNAME	Program specification block ID	8	All variable classes
PST	Partition specification table number	3	All variable classes
REGION	IMS region name	8	Any PERFormance variable

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
STARTTIME	None	8 (*hh:mm:ss* format)	All variable classes
SYSID	z/OS system name	8	All variable classes
TRANID	Transaction ID	8	All variable classes
USERID	User ID	8	All variable classes

ID Options for MQ

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
ENDTIME	None	8 (*hh:mm:ss* format)	Any MQ variable
JOBNAME	The name of the job	8	Any JOB-related class
OBJNAME	MQ object name	48	Any MQ variable
OBJTYPE	MQ object type	8	Any MQ variable
OBJSUBTYPE	MQ object subtype	8	Any MQ variable
QMANAGER	Queue manager name	4	Any MQ variable
RESNAME	Resolved object name	48	Any MQ variable
RESMANAGER	Resolved queue manager name	48	Any MQ variable
STARTTIME	None	8 (*hh:mm:ss* format)	Any MQ variable
SYSID	System ID of the z/OS system	8	Any MQ variable

Note: Use the FORMAT command with the ID option to truncate long identifiers that waste space on the report page.

FORMAT Option

FORMAT Option (see page 158)

ID Options for AUDITLOG

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
STARTTIME	None	8 (*hh:mm:ss* format)	Any AUDIT LOG ALL variable
ENDTIME	None	8 (*hh:mm:ss* format)	Any AUDIT LOG ALL variable
AUDTYPE	Audit record type	16	Any AUDIT LOG ALL variable
SYSTEM	System name	8	Any AUDIT LOG ALL variable
SUBSYS	Subsystem ID	4	Any AUDIT LOG ALL variable
AJOBID	Audit job ID	8	Any AUDIT LOG ALL variable
TASKNAME	Task name	8	Any AUDIT LOG ALL variable
SECGROUP	Security group	8	Any AUDIT LOG ALL variable
RELEASE	Product release	4	Any AUDIT LOG ALL variable
AJOBNAME	Audit job name	8	Any AUDIT LOG ALL variable
ATERMID	Audit terminal ID	8	Any AUDIT LOG ALL variable
AUSERID	Audit user ID	8	Any AUDIT LOG ALL variable

ID Options for COMMON

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
APPLID	Application ID for CICS	8 (*hh:mm:ss*)	Any COMmon EXCeption variable
CJOBID	Job ID for Common variables	8	Any COMmon EXCeption variable
GROUP	Any exception group (System, ASID, Device, CPU, and so on)	12	Any COMmon EXCeption variable
JOBNAME	Job name	8	Any COMmon EXCeption variable
OLDSTATUS	Exception owners. Valid values: MVS, MQSeries, IMS, TCP/IP, CICS	8	Any COMmon EXCeption variable

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
OWNER	Owner (MVS, MQSeries, IMS, TCP/IP, CICS)	8	Any COMmon EXCeption variable
RESRC	Exception resource 1 names	8	Any COMmon EXCeption variable
RESRC2	Exception resource 2 names	8	Any COMmon EXCeption variable
RULETYPE	Threshold rule types. Valid values: None, Upper, Lower, Change	6	COMmon THReshold
STARTTIME	None	8 (*hh:mm:ss*)	Any COMmon EXCeption variable
STATUS	Exception status	8	Any COMmon EXCeption variable
SUBGROUP	Any exception subgroup (SYSTEM, CPU, STORAGE, and so on)	8	Any COMmon EXCeption variable
SYSTEM	System name	8	Any COMmon EXCeption variable
THRECLAS	Threshold classes. Valid values: Count, Cnt K (count in 1024), Rate, Time, Pct	5	Any COMmon EXCeption variable
THRETYPE	Threshold types. Valid values: *, System, Tran-End, State, Tran-Sum, Timer, Tran-Dyn, Tran-Sys	8	COMmon THReshold
TRANID	CICS transaction IDs	8	COMmon THReshold
VARNAME	Exception variable names	8	Any COMmon EXCeption variable

ID Options for MVS

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
AACCTID	Jobs and steps from the JOB statement accounting field	varies	EXCP JOB OPERator non-SYSTEM PAGE PERFormance PROCessor STEP STORage
ASTYPE	The following address space types: JOB, STC (started task), TSO	3	DSN EXCP non-SYSTEM ID JOB OPERator non-SYSTEM PAGE PERFormance PROCessor STEP STORAGE TSO
CHNID	Channel IDs, in hexadecimal	2	CHANNEL
CHNTYPE	Channel type acronym	5	CHANNEL
CMPCODE	Completion code	4	JOB STEP
CPUID	CPU serial numbers of MVS hosts	6	CPU LPAR
CPUPID	CPU processor IDs of MVS processors	2	CPU LPAR
DDNAME	DD names of data sets	8	EXCP
DEVID	Device addresses, in hexadecimal	4	DEVICE DISK DSN EXCP

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
DISKID	Disk addresses, in hexadecimal	4	DISK DSN
DSND	Data set names	44	DSN PAGESWAP
ENDTIME	None; record end times are printed	8	Any variable
FORMID	Output form numbers	4	MVS SYSOUT variables
JOBCLASS	Job classes	1	EXCP JOB OPERator non-SYSTEM PAGE PERFormance PROCessor STEP STORAGE
JOBID	Jobname	8	DSN EXCP non-SYSTEM ID JOB OPERator non-SYSTEM PAGE PERFormance PROCessor STEP STORAGE TSO
LPARID	Names of PR/SM logical partitions	8	LPAR
LPARNAME	Name of the operating system instance	8	CPU
LPID	Logical processor IDs	2	LPAR

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
PROGID	Program names	1-8	STEP EXCP STEP OPERator STEP PAGE STEP PERFormance STEP PROCessor STEP STEP STORage
PROGRAMMER	Programmer names	1-20	EXCP JOB OPERator non-SYSTEM PAGE PERFormance PROCessor STEP STORAGE
SMF6FMN	Output form numbers	4	MVS print variables
SMF6OWC	SYSOUT classes	1	MVS print variables
STARTTIME	None; record start times are printed	8	Any variable
STEPID	Step IDs	8	DSN STEPEXCP non-SYSTEM IO STEP OPERator STEP PAGE STEP PERFormance STEP PROCessor STEP STEP STORage TSO
SWAPID	Swap reasons	30	SWAP
SYSID	System IDs of MVS systems	8	Any MVS variable
TSOCMD	TSO command names	8	TSO

ID Option	Identifiers You Can Specify with the ID Option	Length of Identifiers	Variable Classes You Can Use with the ID Option
USERID	User information	1-8	EXCP JOB OPERator non-SYSTEM PAGE PERFormance PROCessor STEP STORAGE
VOLSER	Volume IDs	6	DSN DISK

ID Options Used with Commands

Use ID options with commands to organize the data in a report by a specific type of resource, such as transaction, terminal, program, or user.

You can use ID options with the TAB, TAB2, GRAF, and FLASHBACK commands. You cannot use ID options with any of the plot commands. You can, however, use ID options with variables in plot reports. For a description, see Generate a Plot Report Using Variables (see page 184) in this chapter.

To use an ID option with a command, specify the ID option after the command and before any variables.

Limitations Caused by Using Identifiers

You are not required to specify identifiers when you use an ID option with a command.

If you specify one or more identifiers with an ID option, the resulting report is:

- Organized by the type of resource specified by the ID option
- Limited to the resources specified by the identifiers

ID Options Syntax

The syntax of ID options when used with commands is as follows:

idoption[(*identifier*[,*identifier*...])]

idoption

Includes one of the ID options listed in ID Options for MVS (see page 174) in this chapter.

The specified ID option determines the type of resource to organize the resulting report by.

identifier

Specifies an individual resource or group of resources to include in the report. The types of identifiers that are valid for each ID option are given beginning in Use ID Options with Variables and Identifiers (see page 181) in this chapter.

More information:

ID Options for IMS (see page 170)
Use ID Options with Variables and Identifiers (see page 181)
Identifiers (see page 186)

Examples of ID Options Used with Commands

The following pages show some examples of using ID options with commands.

- TAB Report Using an ID Option and Identifiers

  The following TAB command includes the ID option CICSID together with the identifiers C1, C2, and C3 to generate a report. The report shows the number and percentage of transactions used by each of three CICS systems:

```
TAB CICSID(C1, C2, C3) TRAN USE              AND,
                       TRAN USE PCT
END
RUN
```

  A report with the following format is generated:

```
CICS        TRAN       TRAN
 ID          USE       %USE
____       _____     _____
C1           297       51.4
C2           188       32.5
C3            93       16.1
____       _____     _____
SUM          578      100.0
____       ____      _____
```

- TAB and TAB2 Report Using ID Options

  The following example uses the ID option TERMID with the TAB command and the ID option TRANID with the TAB2 command. This example creates a report on terminals and the transactions associated with them:

```
TAB  TERMID TRAN USE AND TRAN LIFE
TAB2 TRANID TRAN USE AND TRAN LIFE
END
RUN
```

  A report with the following format is generated:

```
TERM     TRAN     TRAN     TRAN     TRAN     TRAN
ID       USE      LIFE      ID       USE     LIFE
-------- -------- -------- -------- -------- --------
F014     2483     0.361     FXFR     2482     0.361
                            CSGM        1     0.012
                          --------  ------- --------
                           SUMMARY  2483     0.361
                          -------- -------- --------
.
.
.
                          --------       -------- --------
-------- --------  ----------------       -------- --------
SUM        5044     140.0                    8684     87.43
--------  --------  --------    --------  -------- --------
```

- Graph Report Using an ID Option Without Identifiers

  By default, the GRAF command produces a report on the identifiers associated with the specified variable. If you specify an ID option with the GRAF command, the identifiers for that ID option are used in place of the default.

  For example, graph reports on TRANSACTION variables are, by default, organized by transaction ID. The following commands use the ID option JOBID with the variable TRAN USE:

```
GRAF JOBID TRAN USE
END
RUN
```

  These commands produce a report on transaction usage by CICS job rather than by transaction ID:

```
SHIFT  NONE
EACH   DAY                                          TRANSACTION USE
PERIOD NONE
0     191.9     383.8     575.7     767.6     959.5      1151      1343      1535      1727      1919    X
                +.........+.........+.........+.........+.........+.........+.........+.........+ .........+.........+
CICSA           .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  1919
CICSB           .XXXXXXXXXXX       .         .         .         .         .         .         .         .         .   236
CICSTEST        .XXXXXXXXXX        .         .         .         .         .         .         .         .         .   219
CICSX1          .XXXXXX   .        .         .         .         .         .         .         .         .         .   108
CICSX2          .         .        .         .         .         .         .         .         .         .         .     6
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                0     191.9     383.8     575.7     767.6     959.5      1151      1343      1535      1727      1919    X
SYMBOLS:  "X" =  TRANSACTION USE
```

- Graph Report Using an ID Option and Identifiers

  The following GRAF command is specified with the ID option TRANID and the identifiers CMAS, CCOM, and CSSF:

```
GRAF TRANID(CMAS,CCOM,CSSF) TRAN LIFETIME SCALE(1.0)
  FROM 06/06/10 00.00.00
  TO   06/07/10 00.00.00
END
RUN
```

  This command limits the resulting report to data about only those transactions:

```
0     0.100     0.200     0.300     0.400     0.500     0.600     0.700     0.800     0.900     1.000    X
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
CMAS            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  .         .         .         . 0.668
CCOM            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.         .         .         .         .         .         .         . 0.286
CSSF            .XXXXXX   .         .         .         .         .         .         .         .         .         . 0.063
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
0     0.100     0.200     0.300     0.400     0.500     0.600     0.700     0.800     0.900     1.000    X
SYMBOLS:  "X" =  TRANSACTION LIFETIME IN SECONDS.
```

- FLASHBACK Report Using an ID Option and Identifiers

  The following commands produce a FLASHBACK LIST report showing the transactions of operator SWS only:

  ```
  FLASHBACK LIST
  END
  RUN
  ```

  A report with the following format is generated:

```
CICS   END           TERM   TRAN     TRAN      TRAN     TRAN     TERM     TERM     TRAN     FILE     FILE     PROG   ABEND
ID    TIME           ID     ID       NUMBER    LIFE      CPU     RESP      IO       I/O     COUNT     USE     COUNT   CODE
---- -------- ------ ------ -------- -------- -------- -------- -------- -------- -------- -------- ----------------------
B1   00.45.29 0A02   SWS                67    1.011    0.274    1.011        2    0.000        0        0        3
B1   00.45.35 0A02   SWS                69    2.387    0.051    1.187        3    0.199        1        2        6
.
.
.

B1   05.34.14 0A02   SWS               606    0.129    0.032    0.129        2    0.000        0        0        3
---- -------- ------ ------ ---------------- -------- -------- -------- -------- -------- -------- -------- --------------
     05.34.14                         8003    28.47    0.168    0.132      231    0.021        0       10        3
---- -------- ------ ------ ------- -------- -------- -------- -------- -------- ----- ----------- -------- -------------
```

Use ID Options with Variables and Identifiers

Use ID options with individual variables to limit the range of the variable to a particular resource or group of resources. You can use ID options with variables in any type of report. Specify one or more identifiers when you use ID options with variables.

To use an ID option with a variable, specify the ID option and its identifiers immediately following the variable you want to restrict.

Syntax of ID options used with variables:

idoption (*identifier*[,*identifier*...])

idoption

Includes one of the ID options. The ID option you specify determines the type of resource the report is to be about. The ID options that can be used with a given variable depend upon the class and record type. The ID options that are valid for each class and record type are given in the chapter “Variables.”

identifier

Specifies an individual resource or group of resources the variable is limited to.

More information:

ID Options for MVS (see page 174)
Variables (see page 109)

ID Options Used with Commands or Variables

The following illustrates the differences between using ID options with commands and using ID options with variables:

- The first report is organized using both time and CICS ID. The entire report is restricted to data about the CICS systems A1 and B1.
- The second report is organized only using time, and each ID option restricts the range of only the one variable it is specified with. For example, the third column in the report, generated by the variable and ID option TRAN USE CICSID(A1), contains information about system A1 only.

Sample Commands for Report 1

The following commands generate a report showing the number and percentage of transactions used by each of two CICS systems. Specify the ID option and identifiers CICSID(A1,B1) with the TAB command.

```
TAB DATE CICSID(A1,B1) TRAN USE AND TRAN USE PCT
  EACH DAY
  FROM 12/08/10
  TO   12/09/10
END
RUN
```

This command generates the following report:

```
CICS     TRAN   TRAN
DATE      ID     USE   %  USE
-------- ---- -------- --------
12/08/10  B1     368     71.0
          A1     150     29.0
-------- ---- -------- --------
         SUM     518    100.0
-------- ---- -------- --------
12/09/10  B1    5488     79.8
          A1    1393     20.2
-------- ---- -------- --------
         SUM    6881    100.0
-------- ---- -------- --------
```

Sample Commands and Variables for Report 2

The following commands also generate a report showing the number and percentage of transactions used by each CICS system when there are two. However, in this example, the ID option and identifier CICSID(A1) and the ID option and identifier CICSID(B1) are specified with variables.

```
TAB DATE TRAN USE                HDR1(-TOTAL-) HDR2(TRAN USE) AND,
         TRAN USE CICSID(A1)     HDR1(-A1-)    HDR2(TRAN USE) AND,
         TRAN USE CICSID(A1) PCT HDR1(-A1-)    HDR2(% USE)    AND,
         TRAN USE CICSID(B1)     HDR1(-B1-)    HDR2(TRAN USE) AND,
         TRAN USE CICSID(B1)PCT  HDR1(-B1-)   HDR2(% USE)
  EACH DAY
  FROM 12/08/10
  TO   12/09/10
END
RUN
```

This command generates the following report:

```
         -TOTAL-    -A1-     -A1-     -B1-     -B1-
  DATE   TRAN USE TRAN USE  % USE   TRAN USE  % USE
-------- -------- -------- -------- -------- --------
12/08/10    518      150     29.0      368     71.0
12/09/10   6881     1393     20.2     5488     79.8
-------- -------- -------- -------- -------- --------
PERIOD     7399     1543     20.9     5856     79.1
-------- -------- -------- -------- -------- --------
```

Generate a Plot Report Using Variables

The following commands illustrate using an ID option with a variable to generate a plot report that shows the response time for terminal L310:

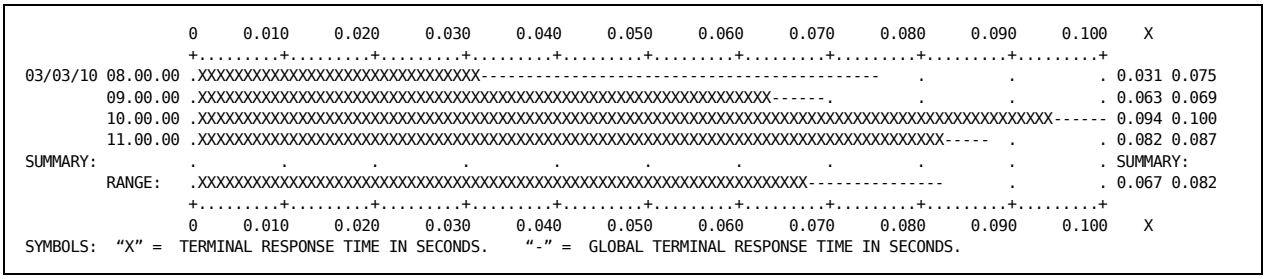
Follow these steps:

1. Enter the following commands: illustrate using an ID option with a variable to generate a plot report that shows the response time for terminal L310:

   ```
   PLOT TERM RESP TERMID(L310)
   END
   RUN
   ```

2. Run the report.

   These commands generate a plot report like the following. This plot contains, with the response time for terminal L310, the response time for all terminals. You can then compare the performance of terminal L310 to the performance of the rest of the terminals in the system.



Generate a Tab Report Using Commands and Variables

A single Report Writer command can include more than one ID option. You can specify an ID option with the command plus specify ID options with one or more of the variables included in the command. Use the following procedure to generate a tab report.

Follow these steps:

1. Specify the TAB command with the ID option TRANID:

   This ID option organizes the report by transaction ID.

2. In addition, specify one of the variables with the ID option and identifier TRANID(CINQ)

   This ID option restricts the range of that variable to the transaction CINQ.

```
TAB TRANID  TRAN USE         AND,
            TRAN LIFETIME    AND,
            TRAN MAXLIFETIME AND,
            TRAN USE TRANID(CINQ) HDR1( TRAN USE ) HDR2(FOR TRAN “CINQ”)
  FROM 03/05/10 08:00:00
  TO   03/05/10 12:00:00
END
RUN
```

3. Run the report

A report similar to the following is generated:

```
TRAN       TRAN     TRAN    MAX TRAN     TRAN USE
 ID         USE   LIFETIME  LIFETIME  FOR TRAN “CINQ”
-------- -------- --------- -------- ----------------
CINQ       1081    0.764     102.5        1081
CMAS        651    1.580     115.2           0
PINQ        651    0.564     53.06           0
IINQ        365    0.443     67.73           0
-------- -------- --------- -------- ----------------
SUMMARY    2748    0.867     102.5        1081
-------- -------- --------- -------- ----------------
```

Note: The last column of the report has only one nonzero entry. Because this report is organized on the ID option TRANID, each row shows information about only the transaction listed in the first column. The last column of the preceding report represents the use of transaction CINQ only. Therefore, only the detail line for transaction CINQ has a nonzero value in the last column.

Identifiers

Identifiers specify particular resources of the type determined by the ID option. The effect of specifying an identifier depends on whether the ID option used with the identifier is specified with a command or with a variable.

- If an ID option is used with a command, specifying identifiers restricts the report to information about the particular resources specified.
- If an ID option is used with a variable, specifying identifiers restricts the range of the variable to the particular resources specified.

Note: To see the identifiers that are valid with each ID option, see the table beginning in ID Options for MVS (see page 174) in this chapter.

Types of Identifiers

Identifiers can specify one of the following:

- An individual resource of the type determined by the ID option.

  For example, an identifier that names a specific user; that is, a user ID, can follow the ID option USERID.

- A group of resources of the type determined by the ID option. You can specify more than one resource in two ways:
  - You can list several individual resources separated by commas.
  - You can specify one or more groups of resources using generic characters, as explained in the following section.

    For example, an identifier that specifies the group of terminals with IDs that begin with the letter C can follow the ID option TERMID.

Generic Characters in Identifiers

Identifiers can contain the following generic characters to report on groups of resources:

- A plus sign (+) tells the Report Writer not to compare that position in the identifier. A place in an identifier filled in with the plus sign (+) is always a match.
- An asterisk (*) tells the Report Writer that all places after and including the asterisk are matches.
- A not sign (¬) tells the Report Writer to exclude the identifier that follows it. You can use the generic characters * and + when specifying the identifier to exclude.

Example: Generic Characters Used in Identifiers

The following GRAF command produces a report on terminal use for only certain transactions:

```
GRAF TRANID(C*,¬C100, S104) TERMINAL USE
```

In this example, the ID option TRANID is specified with three identifiers, C*, ¬C100, and S104. The identifier C* includes all transactions starting with *C*. The identifier ¬C100 excludes the transaction C100. The identifier S104 includes the transaction S104. Therefore, this GRAF command generates a report showing terminal use for transactions starting with *C* except transaction C100, and for the transaction S104.

Specify Identifiers in Hexadecimal

Identifiers can be specified in their hexadecimal representation by placing the hexadecimal value inside single quotes.

The following two commands generate the same report, a report on only those transactions with IDs beginning with *EXP*:

```
TAB TRANID(EXP*) CICS TRAN USE
END
TAB TRANID('C5E7D7'*) CICS TRAN USE
END
```

How Format Identifiers Work

In some situations, you want to change the way identifiers are printed. You can use the following two ways:

- You can use the FORMAT option to limit the number of characters printed for each identifier.
- You can use the DEFINE command to define new identifiers or redefine existing identifiers.

Each of these methods is explained in more detail in the following sections.

Limit the Number of Characters Printed

You can use the FORMAT option to limit the number of characters printed for each identifier. Truncation of the identifier names is done immediately before printing. Therefore, even if two identifiers are identical after truncation, their data will not be combined in the report.

For a complete description of the FORMAT option and its full syntax, see the chapter "Variable Options."

The FORMAT option when used with ID options has the following syntax:

idoption FORMAT(*n*C)

n

Specifies the number of characters to appear in the output. For example, FORMAT(4C) indicates an output of four characters. The value of *n* has no limit.

Examples: With and Without the Format Option

- Sample report without the format option.

  The user IDs in this report could be shortened using the FORMAT option, saving enough column space to include another variable. Suppose your user IDs are eight characters long, a portion of a report on users could look like the following:

USER ID	TRAN USE	TRAN ABENDS	TRAN CPU	TRAN TCPU	TRAN LIFE	TRAN MAX LIFE	TRAN STOR	TRAN WAITS	PROG USE	PROG ABENDS	TERM WRITES
AALBERTS	4	0	0.024	0.097	0.233	0.749	8934	3	16	0	5
AANDERSO	3	0	0.016	0.049	0.110	0.262	11872	3	15	0	4
ABROWN	9	2	0.070	0.630	83.11	370.5	8150	4	51	2	14
ABROWNIN	63	0	0.037	2.321	1.917	19.00	7665	5	198	0	44
ACHRISTI	3	1	0.082	0.247	123.7	370.8	11125	5	18	1	5
AKENNEDY	19	0	0.166	3.150	20.43	347.5	34520	41	61	0	16
ALEWIS	4	0	0.859	3.437	6.840	13.18	116K	13	176	0	4
APETERSO	12	0	0.080	0.961	0.240	0.472	22526	6	66	0	32
BAUSTEN	48	1	0.109	5.245	16.79	715.1	37487	10	304	1	56
.	.	.	.	.	.	.	.	.	.	.	.

- Sample report with the FORMAT option

  The following commands include the option FORMAT(4C) and the additional variable TERM RESP:

```
TAB USERID FORMAT(4C) FOR 10 SORT ALPHA,
TRAN USE    AND TRAN ABENDS    AND,
     TRAN CPU    AND TRAN TCPU      AND,
     TRAN LIFE   AND TRAN MAXLIFE   AND,
     TRAN STOR   AND TRAN WAITS     AND,
     PROGRAM USE AND PROGRAM ABENDS AND,
     TERM USE    AND TERM RESP
END
RUN
```

The following report is produced:

```
USER  TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN    PROG     PROG     TERM     TERM
  ID   USE  ABENDS      CPU     TCPU     LIFE  MAX LIFE     STOR    WAITS     USE  ABENDS   WRITES     RESP
---- ------- -------- -------- -------- -------- -------- -------- ------- ------- -------- -------- ------
AALB     4        0    0.024    0.097    0.233    0.749     8934       3      16        0        5    0.179
AAND     3        0    0.016    0.049    0.110    0.262    11872       3      15        0        4    0.075
ABRO     9        2    0.070    0.630    83.11    370.5     8150       4      51        2       14    0.213
ABRO    63        0    0.037    2.321    1.917    19.00     7665       5     198        0       44    0.344
ACHR     3        1    0.082    0.247    123.7    370.8    11125       5      18        1        5    0.218
AKEN    19        0    0.166    3.150    20.43    347.5    34520      41      61        0       16    1.467
ALEW     4        0    0.859    3.437    6.840    13.18     116K      13     176        0        4    6.840
APET    12        0    0.080    0.961    0.240    0.472    22526       6      66        0       32    0.072
BAUS    48        1    0.109    5.245    16.79    715.1    37487      10     304        1       56    1.499
 .       .    .    .        .     .   .     .   .        .       .      .        .        .        .
 .       .    .    .        .     .   .     .   .        .       .      .        .        .        .
```

Note: In the report, the user ID ABRO appears twice in the first column because the first four characters of two user IDs were the same.

Define New or Redefine Existing Identifiers

Use the DEFINE command to define new identifiers, redefine existing identifiers, or define new variables.

For a complete description of the DEFINE command and its full syntax, see the chapter “Commands.”

Using the DEFINE Command Versus the FORMAT Option

Use care when using the DEFINE command to create shortened forms of identifiers. Unlike the FORMAT option, the DEFINE command can merge data about different identifiers. For instance, the DEFINE command could be used to redefine the ID option USERID to be the first four characters of the user ID. Using this command in the examples With and Without the Format Option instead of the FORMAT option merges the data for ABROWN and ABROWNING. Both user IDs would have been redefined to the same value, ABRO.

However, the DEFINE command offers more flexibility than the FORMAT option.

For example, the FORMAT option lets you truncate identifiers and retain the first characters. However, the DEFINE command lets you select any portion of the identifiers.

For instance, in examples With and Without the Format Option, the ID option USERID could be:

- Redefined to a length of four characters
- Begin with the second character rather than with the first

Chapter 8: Tabular Reports

This section contains the following topics:

About Tabular Reports (see page 191)
TAB Command (see page 192)
Reports Organized by Time and Date or by Resources (see page 192)
Specify Variables (see page 196)
Variable Options (see page 200)
Limit the Contents of a Report (see page 204)
How Report Sorting Works (see page 209)
Use Other Commands with the TAB Command (see page 212)
TAB2 Command (see page 220)

About Tabular Reports

Tabular reports organize data into rows and vertical columns.

Labels or *keys* at the beginning of each row describe what the data in the line represents:

- A time period
- A particular resource, such as a transaction, terminal, or user

Headings at the top of each column describe a variable; the value of that variable for each time period or resource appears in that column.

Be familiar with the following:

- Report Writer commands
- Report Writer variables and lists of the ID options to use with each class of variable
- ID options and their use with commands and variables

More information:

Commands (see page 39)
ID Options (see page 167)
Variable Options (see page 157)

TAB Command

Use the TAB command to produce tabular reports. The variables you specify with the TAB command determine the columns in your report and the order in which they appear.

The operands you specify determine:

- How the report is organized: by date or time, by resource type or individual resources (such as transactions, terminals, or users), or by both
- How the report is sorted
- The column headings that appear in the report

More information:

Commands (see page 39)

Reports Organized by Time and Date or by Resources

You can organize tabular reports by date and time, by a type of resource, or by both. Examples of each of these types of organization are provided in the following sections.

Reports Organized by Date and Time Using Tab Operands

You can use the following TAB command operands to organize reports by date, time, or day of the week.

- DATETIME - Sets the date and time
- DATE - Sets the date
- TIME - Sets the time
- DAY - Sets the day of the week

DATETIME Operand—Organize by Date and Time

Use the DATETIME operand to create tabular reports organized by date and time.

Syntax to create a tabular report displaying transaction use, transaction lifetime, and maximum transaction lifetime:

```
TAB DATETIME TRAN USE AND,
             TRAN LIFE AND,
             TRAN MAXLIFE
 EACH HOUR
 FROM 12/08/08 10:00:00
 TO   12/08/08 18:00:00
END
RUN
```

A report like the following is generated where the date is the same for each row in the report:

```
TRAN     TRAN    TRAN     TRAN
DATE     TIME    USE     LIFE       MAX LIFE
-------- ----- -------- -------- -----------------
12/08/08 10.00      18    213.3    912.3
12/08/08 11.00      63    280.0   2980
12/08/08 12.00      39    515.7   5508
12/08/08 13.00     120    124.9    912.5
12/08/08 14.00      20    729.7    912.8
12/08/08 15.00      20    729.8    912.6
12/08/08 16.00      22    663.4    912.8
12/08/08 17.00      22    580.7    913.0
-------- -----  -------- -------- --------
PERIOD             324    349.2   5508
-------- -----  -------- -------- --------
```

TIME Operand—Organize by Time

Use the TIME operand to create tabular reports organized by time.

To eliminate the date, use the TIME operand instead of the DATETIME operand, as follows:

```
TAB TIME TRAN USE  AND,
         TRAN LIFE AND,
         TRAN MAXLIFE
EACH HOUR
FROM 12/08/08 10:00:00
TO   12/08/08 18:00:00
END
RUN
```

A report like the following is generated:

```
         TRAN      TRAN      TRAN
 TIME     USE      LIFE    MAX LIFE
-----  -------- --------- --------
10.00      18    213.3     912.3
11.00      63    280.0      2980
12.00      39    515.7      5508
13.00     120    124.9     912.5
14.00      20    729.7     912.8
15.00      20    729.8     912.6
16.00      22    663.4     912.8
17.00      22    580.7     913.0
-----  -------- --------- --------
PERIOD    324    349.2      5508
-----  -------- --------- --------
```

DATE Operand—Organize by Date

Use the DATE operand to create tabular reports organized by date.

To create a report organized by date, use the following commands:

```
TAB DATE TRAN USE  AND,
         TRAN LIFE AND,
         TRAN MAXLIFE
END
RUN
```

DAY Operand—Organize by the Day of the Week

Use the DAY operand to create tabular reports organized by the day of the week (Monday, Tuesday, and so on).

To create a report organized by day, use the following commands:

```
TAB DAY TRAN USE    AND,

        TRAN ABENDS AND,
        TRAN CPU    AND.
        TRAN LIFE   AND,
        TRAN MAXLIFE
END
RUN
```

A report like the following is generated:

```
            TRAN      TRAN      TRAN      TRAN      TRAN
 WEEKDAY     USE     ABENDS      CPU      LIFE    MAX LIFE
--------- -------- -------- -------- -------- --------
FRIDAY        59        1    25.29    778.8     4529
SATURDAY    1226       60    0.208    221.3    43995
SUNDAY       518        3    0.392    638.9    46487
MONDAY      6881       77    0.515    16.13     4036
--------- -------- -------- -------- -------- --------
PERIOD      8684      141    0.633    87.43    46487
--------- -------- -------- -------- -------- --------
```

Organize a Report by Resource Using an ID Option

You can organize tabular reports by a type of resource, such as transactions or terminals, or by individual resources within a type. Specify an ID option with or without identifiers following the TAB command and preceding any variables. Identifiers are listed in the first column, in descending order of the value of the first variable specified. For more about sorting reports, see How Report Sorting Works (see page 209) in this chapter.

The following example uses the same commands as in the example DATETIME Operand except that the ID option TRANID replaces the DATETIME operand. Using TRANID organizes the resulting report by transaction rather than by date and time.

```
TAB TRANID TRAN USE      AND,
           TRAN LIFE     AND,
           TRAN MAXLIFE
 EACH HOUR
 FROM 12/08/08 10:00:00
 TO   12/08/08 18:00:00
END
RUN
```

A report like the following is generated:

```
TRAN         TRAN      TRAN      TRAN
ID           USE      LIFE    MAX LIFE
-------- -------- -------- --------
PSPI          97    0.149     1.060
TCP           57    912.2     913.0
.
.
.
VGBR           1    0.371     0.371
-------- -------- -------- --------
SUMMARY      324    349.2      5508
-------- -------- -------- --------
```

Organize by Date, Time, or Both and Resource

You can specify both an ID option and either the DATETIME, DATE, TIME, or DAY operand to create a more detailed tabular report.

The following commands generate a report organized by both time and transaction. The FOR operand limits the report to the five most active transactions during each interval.

```
TAB TIME TRANID FOR 5,
    TRAN USE  AND,
    TERM RESP AND,
    DATASET USE
   EACH HOUR
   FROM 12/08/08 12:00:00
   TO   12/08/08 15:00:00
END
RUN
```

More information:

FOR Operand—Limit the Identifiers Included in a Report (see page 205)

Specify Variables

You can specify multiple variables with the TAB command. The resulting report includes a column of data for each variable specified, in the order in which the variables are specified.

ID Options Specified with the TAB Command

If you do not specify an ID option with the TAB command, there are no restrictions on the variables included in the command.

If you specify an ID option with the TAB command, the type of resource associated with the ID option organizes the resulting report. Accordingly, you can then use only variables that apply to that type of resource. For example, you cannot create reports showing transaction activity by data set, program, or transient data.

However, you can use variables of different classes within a report organized by a type of resource. A report organized by terminal, for example, can include PROGRAM or TRANSACTION variables (among others), because these variables apply to terminal activity.

More information:

ID Options (see page 167)

Include CICS and MVS Variables

You can produce tabular reports containing data on both CICS and MVS using any combination of CICS and MVS variables. Review the restrictions described in the previous section.

Use the following commands to generate a report using data from both CICS and MVS:

```
INPUT(TAPEIN)
 TAB DATE MVS DISK SCH HDR1(DISK) HDR2(SIO COUNT) AND,
        MVS DISK SCH RATE   HDR1(DISK) HDR2(SIO RATE)  AND,
        MVS DISK SERVTIME                              AND,
        MVS DISK SERVTIME MAX HDR1(MAX) HDR2(TIME)     AND,
        TRAN USE                                       AND,
        TRAN LIFE
   EACH 1 DAY
   FROM 01/16/08
END
RUN
```

A report like the following is generated:

```
   DISK      DISK    SERVICE    MAX      TRAN     TRAN
  DATE   SIO COUNT  SIO RATE   TIME   SERVTIME    USE   LIFETIME
-------- ---------- -------- -------- -------- -------- --------
01/16/08      2,156   0.02    0.023    0.093      154    1.218
01/17/08    170,270   1.97    0.023    0.244      461    0.822
01/18/08     74,139   2.42    0.018    0.069      223    0.761
-------- ---------- -------- -------- -------- -------- --------
 PERIOD     246,565   0.53    0.022    0.244      838    0.878
-------- ---------- -------- -------- -------- -------- --------
```

Use ID Options with Variables

Use of ID options with variables lets you limit the range of the variable to a specific resource or group of resources.

To limit the range of a variable, specify an ID option and one or more identifiers following the variable. The use of ID options with variables is explained in detail in the chapter “ID Options.”

Note: Not all ID options can be used with all variables. For information about which ID options you can use with each variable class, see the description of the variable class in the chapter “Variables.”

This example uses the ID option JOBNAME to generate a single report containing information about two systems:

```
EACH 1 DAY
TAB DATE TRAN USE                    HDR1(-TOTAL-) HDR2(TRAN USE) AND,
         TERM RESP                   HDR1(-TOTAL-) HDR2(TERMRESP) AND,
         TRAN USE   JOBNAME(CICS1)   HDR1(-CICS1-) HDR2(TRAN USE) AND,
         TRAN USE  JOBNAME(CICS1) PCT HDR1(-CICS1-) HDR2(%USE)     AND,
         TERM RESP JOBNAME(CICS1)    HDR1(-CICS1-) HDR2(TERMRESP) AND,
         TRAN USE  JOBNAME(CICS2)    HDR1(-CICS2-) HDR2(TRAN USE) AND,
         TRAN USE  JOBNAME(CICS2) PCT HDR1(-CICS2-) HDR2(%USE)     AND,
         TERM RESP JOBNAME(CICS2)    HDR1(-CICS2-) HDR2(TERMRESP)
 END
RUN
```

A report like the following is generated:

```
-TOTAL-  -TOTAL-  -CICS1-  -CICS1-  -CICS1-  -CICS2-  -CICS2-  -CICS2-
DATE    TRAN USE TERMRESP TRAN USE   %USE   TERMRESP TRAN USE   %USE   TERMRESP
--------  -------- -------- -------- -------- -------- -------- -------- ------
02/01/08    1134    2.342      761     67.1    1.445      307     27.1    4.844
02/08/08    2067    1.047      693     33.5    0.762      723     35.0    1.221
02/08/08    1893    1.758      730     38.6    1.942      463     24.5    2.008
02/04/08    1872    1.039      790     42.2    1.126      484     25.9    0.901
02/05/08    1061    0.772      665     62.7    0.794      143     13.5    1.066
02/06/08    1297    1.526      350     27.0    1.420      271     20.9    3.054
--------  -------- -------- -------- -------- -------- -------- -------- ------
 PERIOD     9324    1.602     3989     39.8    1.443     2391     24.7    2.280
--------  -------- -------- -------- -------- -------- -------- -------- ------
```

Changing Column Headings

Each variable is assigned an abbreviation that is used by default as the heading for columns containing data about the variable. You can use the HDR1 and HDR2 operands to replace the top and bottom lines of the heading, respectively.

Changing headings is especially useful when you:

- Change the type of data that a variable represents (for example, from a count to a rate). For an example, see the example in RATE OPTION-Display a Rate Instead of a Count (see page 204) in this chapter.
- Use an ID option and identifiers to restrict the scope of a variable to particular resources, as shown in the following example.
- Shorten the column headings to gain space.

You have no restrictions on the characters used in a heading.

Sample Commands

The variable TRAN USE is specified twice in the following example. By default, the column created for each occurrence of this variable would have the heading *TRAN USE*.

The second occurrence of the variable applies only to the transaction PSPI. Therefore, including the HDR1 and HDR2 operands create the heading *TRAN USE FOR TRAN "PSPI"* above the column for this variable.

```
TAB TIME TRAN USE     AND.
         TRAN LIFE    AND,
         TRAN MAXLIFE AND,
         TRAN USE TRANID(PSPI) HDR1(TRAN USE) HDR2(FOR TRAN "PSPI")
 EACH HOUR
 FROM 12/08/08 12:00:00
 TO   12/08/08 16:00:00
END
RUN
```

A report like the following is generated:

```
TRAN      TRAN      TRAN     TRAN USE
TIME       USE      LIFE        MAX     LIFE FOR TRAN "PSPI"
-----  --------- --------- --------- ---------------
12.00       39     515.7      5508             8
13.00      120     124.9     912.5            84
14.00       20     729.7     912.8             0
15.00       20     729.8     912.6             0
----- --------- --------- --------- ---------------
PERIOD     101     323.0      5508            92
----- --------- --------- --------- ---------------
```

Variable Options

You can use any of the variable options described in the chapter "Variable Options" with the TAB command. Also, following four variable options can be used only with the TAB command. Note the links where you can find examples for using these options.

DIFF

Use this option in a tabular report organized by time or date. This option creates a column displaying the percentage of change in the variable value between time intervals. See DIFF Option-Percentage of Change Between Time Intervals (see page 201) in this chapter.

RANGE

Use this option for a report to test range criteria that is applied after all data is read and before a report is printed. For an example, see Use the RANGE Option (see page 215) in this chapter. Also see the RANGE command in How the RANGE Command Specifies Ranges (see page 212) in this chapter.

RIGHT

Use this option to shift the column for a variable to the right. For an example, see RIGHT Option-Shift the Column for a Variable to the Right (see page 203) in this chapter.

SELECT

Use this option to apply to a variable the period specified with an INCLUDE or EXCLUDE command. For an example, see SELECT Option-Specify a Time Period for a Variable (see page 202) in this chapter.

An example using the RATE option is shown in RATE OPTION-Display a Rate Instead of a Count (see page 204) in this chapter. This option is used with a variable that usually displays a count, but in this instance, displays the rate for the variable instead.

DIFF Option—Percentage of Change Between Time Intervals

You can organize tabular reports by time or date. Use the DIFF option to create a column displaying the percentage of change in the value of a variable between each time interval. The percentage of change is calculated with the following formula:

```
((current value - previous value)/previous value)*100
```

DIFF command has the following syntax:

```
DIFF
```

The following commands use the DIFF option to display the percentage of change of transaction usage. The HDR2 option assigns an appropriate column heading.

```
EACH 1 DAY
TAB DATE TRAN USE AND,
 TRAN USE DIFF HDR2(%CHANGE)
 END
RUN
```

A report like the following is generated:

```
              TRAN         TRAN
   DATE       USE         %CHANGE
--------   ----------   ----------
01/08/08    15,765
01/08/08    22,165      +40.5%
01/04/08    66,374     +199.4%
01/05/08    91,302      +37.5%
01/06/08     2,535      -97.2%
01/07/08     2,123      -16.2%
--------   ---------    ----------
 PERIOD    200,264
--------   ---------    ----------
```

The difference between the number of transactions executed in the first interval and the second interval is 6400 (22,165 - 15,765). The difference is a 40.5 percent increase (6400 / 15,765 = 40.5 percent).

RANGE Option—Test Range Criteria

Use the RANGE option for a report to test range criteria that is applied after all data is read and before a report is printed. The RANGE variable option is best explained with the RANGE DEFERRED command, which has the same result. For an example, see Use the RANGE Option (see page 215) in this chapter.

SELECT Option—Specify a Time Period for a Variable

Use the SELECT option to apply to a particular variable the period specified with an INCLUDE or EXCLUDE command.

The SELECT option has the following syntax:

SELECT(*name*)

name

Must match the *name* operand of the ID operand for the INCLUDE or EXCLUDE command that specifies the time period to apply to the variable.

Example: Sample Commands

The following example assigns the values of *1STSHIFT, 2NDSHIFT,* and *3RDSHIFT* to the *name* operands of the INCLUDE and SELECT commands. Each associated time period from the INCLUDE command is applied to the variable.

```
INCLUDE 08.00.00 - 15.59.59 ID 1STSHIFT
INCLUDE 16.00.00 - 23.59.59 ID 2NDSHIFT
INCLUDE 00.00.00 - 07.59.59 ID 3RDSHIFT
EACH 1 DAY
TAB DAY DATE TRAN USE                    HDR1(TOTAL)     HDR2(USAGE)    AND,
             TRAN USE  SELECT(1STSHIFT) HDR1(1ST SHIFT) HDR2(USAGE)    AND,
             TRAN USE  SELECT(2NDSHIFT) HDR1(2ND SHIFT) HDR2(USAGE)    AND,
             TRAN USE  SELECT(3RDSHIFT) HDR1(3RD SHIFT) HDR2(USAGE)    AND,
             TRAN LIFE                  HDR1(AVG)       HDR2(LIFETIME) AND,
             TRAN LIFE SELECT(1STSHIFT) HDR1(1ST SHIFT) HDR2(LIFETIME) AND,
             TRAN LIFE SELECT(2NDSHIFT) HDR1(2ND SHIFT) HDR2(LIFETIME) AND,
             TRAN LIFE SELECT(3RDSHIFT) HDR1(3RD SHIFT) HDR2(LIFETIME)
 END
RUN
```

A report like the following is generated:

```
          TOTAL  1ST SHIFT 2ND SHIFT 3RD SHIFT   AVG    1ST SHIFT 2ND SHIFT 3RD SHIFT
 WEEKDAY    DATE   USAGE   USAGE  USAGE  USAGE  LIFETIME LIFETIME  LIFETIME  LIFETIME
--------- -------- ------- ------ ------ ------ -------- --------- --------- --------
Friday    03/12/08   962     623    270     69   139.2     89.60     214.9     290.8
Saturday  03/13/08   417     199     18    200   288.7     290.3     270.3     288.8
Sunday    03/14/08    42              42          279.3               279.3
Monday    03/15/08   365     175           190   287.4     272.9               300.8
--------- -------- ------- ------ ------ ------ -------- --------- --------- --------
PERIOD              1786     997    330    459   207.7     161.8     226.1     294.1
--------- -------- ------- ------ ------ ------ -------- --------- ---------   ------
```

RIGHT Option—Shift the Column for a Variable to the Right

Use the RIGHT option to shift the column for a variable to the right. The entire column, including the headings, is shifted.

The RIGHT option has the following syntax:

```
RIGHT(n)
```

n

Indicates the number of print columns (spaces) to shift the column to the right.

Example: Shifting Columns

This example specifies the RIGHT variable option with the variable TRAN USE to shift the column to the right. The column appears under the first heading specified with the HEADER command.

```
HEADER(20,24) ---------TOTAL----------
HEADER(47,26) -------TRAN CSSN--------
TAB DAY TRAN USE RIGHT(8)      AND,
        TRAN LIFE              AND,
        TRAN CPU               AND,
        TRAN USE  TRANID(CSSN) AND,
        TRAN LIFE TRANID(CSSN) AND,
        TRAN CPU  TRANID(CSSN)
 END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY
PERIOD NONE        ---------TOTAL----------   -------TRAN CSSN--------
                     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN
 WEEKDAY              USE     LIFE      CPU      USE     LIFE      CPU
---------          -------- -------- -------- -------- -------- --------
Thursday             1258    205.1    0.032      129    8.756    0.010
Friday                369    175.3    0.028       66    1.773    0.011
---------          -------- -------- -------- -------- -------- --------
PERIOD               1627    198.3    0.031      195    6.393    0.011
---------          -------- -------- -------- -------- -------- --------
```

RATE OPTION—Display a Rate Instead of a Count

Use the RATE option with a variable that usually displays a count to display the rate for that variable.

Example: Sample Commands

The following commands produce a report that displays the transaction usage of all active transactions by transaction ID. The RATE option is used to give the transaction rate (number per second) also.

```
EACH 1 DAY
TAB DATE TRAN USE      TRANID(CINQ) HDR1(CINQ) HDR2(USE)      AND,
         TRAN USE DIFF TRANID(CINQ) HDR1(CINQ) HDR2(%CHANGE)  AND,
         TRAN USE RATE TRANID(CINQ) HDR1(CINQ) HDR2(RATE)     AND,
         TRAN USE      TRANID(PINQ) HDR1(PINQ) HDR2(USE)      AND,
         TRAN USE DIFF TRANID(PINQ) HDR1(PINQ) HDR2(%CHANGE)  AND,
         TRAN USE RATE TRANID(PINQ) HDR1(PINQ) HDR2(RATE)
 END
RUN
```

A report like the following is generated:

```
-          CINQ      CINQ   CINQ     PINQ     PINQ   PINQ
- DATE      USE     %CHANGE  RATE      USE    %CHANGE  RATE
-------- -------- -------- ------ -------- -------- ------
01/08/08    2,978          0.03    12,778           0.15
01/08/08    5,304   +78.1%  0.06    16,861   +31.9%  0.20
01/04/08   30,734  +479.4%  0.36    35,452  +110.2%  0.41
01/05/08   38,711   +25.9%  0.45    52,322   +47.5%  0.61
01/06/08    2,113   -94.5%  0.02       395   -99.2%  0.00
01/07/08      417   -80.2%  0.01     1,610  +307.5%  0.02
-------- -------- -------- ------ -------- -------- ------
 PERIOD    80,257           0.16   119,418           0.24
-------- -------- -------- ------ -------- -------- ------
```

This report shows that in the 86,400 seconds on 01/08/08, transaction CINQ was executed 2978 times, which is a rate of 0.03 times per second.

Limit the Contents of a Report

The following sections describe different ways of limiting the contents of reports.

Limit the Resources Included in a Report

Limit a report to a subset of a resource type by specifying one or more identifiers with an ID option used with the TAB command.

Example: Sample Commands

The following commands include the ID option TERMID and the identifier 0A++ to display data about all 0A terminals:

```
TAB TERMID(0A++)  TERM RESP    AND,
                  TERM I/OTIME AND,
                  TRAN USE
END
RUN
```

FOR Operand—Limit the Identifiers Included in a Report

Use the FOR operand to limit the number of identifiers included in a report. Add the operand FOR 5 to the commands used in the preceding example. Using the operand FOR 5 limits the report to only those five 0A terminals with the highest response times, as follows:

```
TAB TERMID(0A++) FOR 5 TERM RESP    AND,
                       TERM I/OTIME AND,
                       TRAN USE
END
RUN
```

A report like the following is generated:

```
  TERM     TERM     TRAN     TRAN
   ID      RESP     TMIO      USE
-------- -------- -------- --------
0A0E      0.793    2.686      59
0A12      0.226    1.477     401
0A14      0.210    4.747      31
0A19      0.202    144.2      80
0A02      0.194    10.71     115
-------- -------- -------- --------
SUMMARY   0.220    19.92      686
-------- -------- -------- --------
```

FOR Operand—Limit Reports Organized by Time and Date

To limit reports organized by time and date, use the FOR operand. The time periods included in the report contain the highest value for the variable that the FOR operand is specified with.

The following commands produce a report limited to the five times with the highest values for the variable:

```
TAB DATETIME TRAN USE FOR 5
    EACH 1 HOUR
END
RUN
```

A report like the following is generated:

```
                   TRAN
  DATE    TIME      USE
-------- -----  --------
02/22/08 03.00      42
02/22/08 04.00      19
02/22/08 06.00      29
02/22/08 11.00      16
02/22/08 15.00      20
-------- -----  --------
PERIOD              126
-------- -----  --------
```

TOP and BOTTOM Operands—Limit Reports

The TOP and BOTTOM operands, like the FOR operand, limit reports organized by identifier to the number of identifiers specified.

However, the TOP and BOTTOM operands save storage because:

- The FOR operand tests identifiers only after all the data has been read and processed.
- The TOP and BOTTOM operands cause the Report Writer to test each record as it is read and discard records that do not qualify.

The TOP and BOTTOM operands are appropriate for ID options with many unique identifiers, such as TRANNUM.

Example: Sample Commands

- The following commands create a report on the five transactions with the highest terminal response times. Processing similar commands using the FOR operand instead of the TOP operand would require much more storage.

```
TAB TRANNUM TOP 5 TERM RESP AND TRAN TCPU
END
RUN
```

A report like the following is generated:

```
  TRAN      TERM      TRAN
 NUMBER     RESP      TCPU
-------- -------- --------
0005671   38.82    5.740
0000276   28.19    0.007
0000794   24.26    0.059
0000027   21.65    0.065
0001157   16.20    0.654
-------- -------- --------
SUMMARY   25.56    6.525
-------- -------- --------
```

- The following commands create a report on the 30 transactions with the shortest lifetimes. The FORMAT variable option is used to increase the number of digits in the output.

```
TAB TRANNUM TRAN LIFE FORMAT(N.NNNNNNN) BOTTOM 30
END
RUN
```

A report like the following is generated:

```
  TRAN         TRAN
 NUMBER       LIFE
--------    ---------
0000080     0.0056320
0000037     0.0058880
0000775     0.0061440
.
.
.
0000102     0.0076800
--------    ---------
SUMMARY     0.0071680
--------    ---------
```

How Report Sorting Works

Sorting reports lets you customize the resulting report.

To sort reports, consider the following default sorting process:

1. Tabular reports are sorted from left to right with respect to the variable values, in descending order.
2. The leftmost variable is considered first; if the value is the same for two entries, the variable to the right is considered.
3. This sorting continues until variable values differ.

Example: Commands to Sort a Report

```
TAB TERMID(0A++) FOR 5 TERM RESP    AND,
                       TERM I/OTIME AND,
                       TRAN USE
END
RUN
```

This example produces the following report. The terminal 0A0E had the highest response time, so it is listed first. Terminals 0A14 and 0A12 have the same response time, but 0A14 had a higher I/O time, so it precedes 0A12 in the report:

```
TERM      TERM     TRAN     TRAN
 ID       RESP     TMIO      USE
-------- -------- -------- --------
0A0E      0.793    2.686       59
0A14      0.226    4.747       31
0A12      0.226    1.477      401
0A19      0.202    144.2       80
0A02      0.194    10.71      115
-------- -------- -------- --------
SUMMARY   0.220    19.92      686
-------- -------- -------- --------
```

Sort Reports by Different Keys

To sort columns by different keys, alter the order of the columns.

To create a report showing the five terminals using the most transactions, make the variable TRAN USE the first variable in the TAB command:

```
TAB TERMID(0A++) FOR 5 TRAN USE     AND,
                       TERM RESP    AND,
                       TERM I/OTIME
END
RUN
```

A report like the following is generated:

```
   TERM     TRAN     TERM     TRAN
   ID       USE      RESP     TMIO
-------- -------- -------- --------
0A12         401    0.226    1.477
0A09         180    0.064    8.602
0A02         115    0.194    10.71
0A19          80    0.202    144.2
0A0E          59    0.793    2.686
-------- -------- -------- --------
SUMMARY      835    0.197    18.05
-------- -------- -------- --------
```

SORT ASCENDING Operand—Sort in Ascending Order

To sort reports in ascending order, add the SORT ASCENDING operand to the TAB command.

Example: Sample Commands and Reports

- The following commands produce a report sorted so that the five terminals using the fewest transactions are listed in ascending order of transaction use:

```
TAB TERMID(0A++) FOR 5 SORT ASCENDING,
    TRAN USE AND,
    TERM RESP AND,
    TERM I/OTIME
END
RUN
```

  A report like the following is generated:

```
  TERM     TRAN     TERM     TRAN
   ID       USE     RESP     TMIO
-------- -------- -------- --------
0A04           3    0.038    0.004
0A18           6    0.103    0.002
0A0F           6    0.172    34.89
0A11          13    0.158    0.719
0A08          23    0.082    1.970
-------- -------- -------- --------
SUMMARY       51    0.109    5.177
-------- -------- -------- --------
```

- You can specify the SORT operand with a variable other than the first variable. For example, the following commands generate a report sorted by transaction use, the third variable specified:

```
TAB TRANID TRAN TCPU AND,
           TRAN USE SORT ASCENDING
END
RUN
```

A report like the following is generated:

```
   TRAN      TRAN      TRAN
   ID        CPU       USE
-------- --------- --------
RAPP       0.098        2
PINV       1.486        2
PADI       0.350        2
SPAM       0.132        3
RAPL       0.011        4
CSSN       0.071        8
CSGM       0.007        8
CATD       0.105        9
JJJ        0.221       10
CSPQ       0.113       10
VGBC       0.055       13
POSS       0.035       37
PSPI       0.097       43
KCP        7.581       45
TCP        0.572       46
CSNE       0.010       49
X25T       0.132       57
-------- --------- --------
SUMMARY    1.123      348
-------- --------- --------
```

Sort Reports by Identifier

To have your reports sorted by the identifier, add the SORT ALPHA operand to the TAB command.

The following commands generate a report sorted alphanumerically by transaction ID:

```
TAB TRANID FOR 20 SORT ALPHA,
 TRAN LIFE AND TRAN USE AND TRAN MAXLIFE
END
RUN
```

A report like the following is generated:

```
  TRAN      TRAN      TRAN      TRAN
   ID       LIFE       USE   MAX LIFE
-------- --------- -------- ---------
BOAG       14.22        4    27.86
BOEF       35.32        2    58.14
BOEN       44.51        8    130.6
CATD       0.465      125    7.752
CEMT       60.92        4    218.6
CHAI       0.193      160    1.475
CHAT       0.190       24    0.802
CIRB       9.981        2    13.30
CRSQ       0.795        7    3.738
CSAC       0.066       25    0.647
CSDA       0.510        2    1.004


-------- --------- -------- ---------
SUMMARY    77.15     1342    0.000
-------- --------- -------- ---------
```

Use Other Commands with the TAB Command

You can use the secondary commands to tailor tabular reports. For a description, see the chapter “Commands.” Examples of the use of the following commands in tabular reports are provided in the following sections:

- RANGE

  Note: You can use either the RANGE command or the RANGE option to restrict a report to resources whose activity falls within a specified range. For a more detailed explanation of the RANGE command, see the chapter “Commands.” Examples of its use in tabular reports are given in the following sections. For a description of the RANGE option, which is valid only for tabular reports, see Use the RANGE Option (see page 215) in this chapter.

- GROUP
- COST

How the RANGE Command Specifies Ranges

The RANGE command tests range criteria either at the record level (immediate) or, by specifying the DEFERRED operand, before printing the report (deferred).

The RANGE command uses the following process to specify ranges:

1. Tests immediate range criteria before the record is processed.
2. If the range criteria are not met, discards the record for that particular report.

3. Deferred range criteria are not tested until all data is read.
4. Reviews the range criteria before each row is printed. If any variable in the row fails to meet its range criteria, the entire row is discarded.

Example 1: Immediate Range Criteria

This example produces a report about terminals. The RANGE command is specified without the DEFERRED operand to exclude any individual record for which the value of TERM INPUTS is greater than 50.

```
TAB TERMID(0A++),
    TERM INPUTS  AND,
    TERM RESP    AND,
    TERM I/OTIME AND,
    TRAN USE
RANGE TERM INPUTS 0-50
END
RUN
```

This example generates the following report:

```
  TERM     TERM     TERM     TRAN     TRAN
   ID      READS    RESP     TMIO      USE
-------- -------- -------- -------- --------
0A12        401    0.226    1.477      401
0A19        188    0.107    48.08       76
0A09        166    0.064    8.602      180
0A02        143    0.216    4.757      114
0A0E         57    0.793    2.686       59
0A0B         37    0.066    2.362       33
0A14         33    0.210    4.747       31
0A05         25    0.036    0.929       26
0A08         23    0.082    1.970       23
0A11         13    0.158    0.719       13
0A0F         10    0.172    34.89        6
0A18          5    0.103    0.002        6
0A04          2    0.038    0.004        3
-------- -------- -------- -------- --------
SUMMARY    1103    0.168    7.219      971
-------- -------- -------- -------- --------
```

Example 2: Deferred Range Criteria

In the preceding example, several terminals with a total number of inputs (“TERM READS”) not in the specified range are included in the report. Because the RANGE command was specified without the DEFERRED operand, the test of the number of inputs was applied to each individual record. The test was not applied to the total number of inputs for each terminal. Although individual records with more than 50 inputs were excluded, the total for a given terminal can exceed the range.

If instead of the preceding report you want a report on the terminals that had a total of fewer than 50 inputs. You would use the same commands as in the previous example but specify the DEFERRED operand with the RANGE command. The Report Writer applies the range criteria only after all the input data is read. The Report Writer can then determine whether each terminal had a total of fewer than 50 inputs.

The following sample uses the DEFERRED operand with the RANGE command:

```
TAB TERMID(0A++),
    TERM INPUTS  AND,
    TERM RESP    AND,
    TERM I/OTIME AND,
    TRAN USE
RANGE TERM INPUTS 0-50 DEFERRED
END
RUN
```

This example generates the following report:

```
   TERM      TERM     TERM      TRAN      TRAN
   ID       READS     RESP      TMIO       USE
-------- -------- -------- -------- --------
0A0B          37    0.066    2.362        33
0A14          33    0.210    4.747        31
0A05          25    0.036    0.929        26
0A08          23    0.082    1.970        23
0A11          13    0.158    0.719        13
0A0F          10    0.172    34.89         6
0A18           5    0.103    0.002         6
0A04           2    0.038    0.004         3
-------- -------- -------- -------- --------
SUMMARY     148    0.107    3.640       141
-------- -------- -------- -------- --------
```

Use the RANGE Option

The RANGE option yields the same results as the RANGE command with the DEFERRED operand.

The following commands generate the same report as the preceding commands but include the RANGE option instead of the RANGE command:

```
TAB TERMID(0A++),
TERM INPUTS RANGE(0,50) AND,
 TERM RESP              AND,
 TERM I/OTIME           AND,
 TRAN USE
END
RUN
```

Define Groups of Resources Using the Group Command

Use the GROUP command to define groups of resources you want a report to be about.

Example: Sample Group Commands

This example uses the GROUP command to define groups by terminal ID with four divisions defined. Because each group name begins with *CO*, the groups can be selected in the tabular report by specifying TERMID(CO*).

```
GROUP TERMID CO.NORTHERN.DIVISION(L7*)
GROUP TERMID CO.SOUTHERN.DIVISION(L4*)
GROUP TERMID CO.EASTERN.DIVISION(L2*)
GROUP TERMID CO.WESTERN.DIVISION(L1*)
TAB TERMID(CO*) HDR1(DEPARTMENT)  HDR2(NAME),
                  TRAN USE                    AND,
                  TRAN USE RATE     HDR2(RATE)  AND,
                  TRAN LIFE                   AND,
                  TRAN CPU                    AND,
                  TERM RESP                   AND,
                  FILE USE
END
RUN
```

A report like the following is generated:

```
      DEPARTMENT          TRAN      TRAN      TRAN      TRAN      TERM      FILE
        NAME               USE      RATE      LIFE       CPU      RESP       USE
-------------------- --------- --------- --------- --------- --------- ---------
CO.WESTERN.DIVISION     3472      0.02     4.599     0.094     0.360     16925
CO.NORTHERN.DIVISION     976      0.01     16.52     0.120     0.189      1080
CO.SOUTHERN.DIVISION     728      0.00     4.770     0.073     0.106       289
CO.EASTERN.DIVISION      667      0.00     3.688     0.180     0.637      8032
-------------------- --------- --------- --------- --------- --------- ---------
SUMMARY                 5843      0.03     6.508     0.106     0.290     26326
-------------------- --------- --------- --------- --------- --------- ---------
```

Define Groups by Values Within Specified Ranges

You can use the GROUP and RANGE commands together to produce a report on groups of resources whose activity falls within different ranges.

Example: GROUP and RANGE Commands

The following example uses both the GROUP and RANGE commands to define 13 groups of transactions in terms of their lifetimes:

```
 GROUP TRANID RANGE:00.000-01.000(*)  RANGE(TLIFE1)
 GROUP TRANID RANGE:01.001-02.000(*)  RANGE(TLIFE2)
 GROUP TRANID RANGE:02.001-03.000(*)  RANGE(TLIFE3)
 GROUP TRANID RANGE:03.001-04.000(*)  RANGE(TLIFE4)
 GROUP TRANID RANGE:04.001-05.000(*)  RANGE(TLIFE5)
 GROUP TRANID RANGE:05.001-06.000(*)  RANGE(TLIFE6)
 GROUP TRANID RANGE:06.001-07.000(*)  RANGE(TLIFE7)
 GROUP TRANID RANGE:07.001-08.000(*)  RANGE(TLIFE8)
 GROUP TRANID RANGE:08.001-09.000(*)  RANGE(TLIFE9)
 GROUP TRANID RANGE:09.001-10.000(*)  RANGE(TLIFE10)
 GROUP TRANID RANGE:10.001-20.000(*)  RANGE(TLIFE20)
 GROUP TRANID RANGE:20.001-50.000(*)  RANGE(TLIFE50)
 GROUP TRANID RANGE:50.001+(*)        RANGE(TLIFE50+)
 RANGE TRAN LIFE 0-1      ID TLIFE1
 RANGE TRAN LIFE 1-2      ID TLIFE2
 RANGE TRAN LIFE 2-3      ID TLIFE3
 RANGE TRAN LIFE 3-4      ID TLIFE4
 RANGE TRAN LIFE 4-5      ID TLIFE5
 RANGE TRAN LIFE 5-6      ID TLIFE6
 RANGE TRAN LIFE 6-7      ID TLIFE7
 RANGE TRAN LIFE 7-8      ID TLIFE8
 RANGE TRAN LIFE 8-9      ID TLIFE9
 RANGE TRAN LIFE 9-10     ID TLIFE10
 RANGE TRAN LIFE 10-20    ID TLIFE20
 RANGE TRAN LIFE 20-50    ID TLIFE50
 RANGE TRAN LIFE 50-99999 ID TLIFE50+
TAB TRANID(RANGE*) HDR1(TRAN LIFETIME) HDR2(RANGES),
  SORT ALPHA,
                   TRAN USE      AND,
                   TRAN USE PCT  AND,
                   TRAN LIFE     AND,
                   TRAN MAXLIFE  AND,
                   TERM RESP     AND,
                   TRAN CPU      AND,
                   TRAN I/OTIME  AND,
                   TRAN USERTIME AND,
                   TRAN CICSTIME
END
RUN
```

A report like the following is generated:

```
   TRAN LIFETIME       TRAN     TRAN     TRAN     TRAN     TERM     TRAN     TRAN     TRAN     TRAN
       RANGES           USE   %  USE     LIFE   MAX LIFE   RESP      CPU      I/O     USER     CICS
------------------- -------- -------- -------- -------- -------- -------- -------- -------- ------
RANGE:00.000-01.000   6981     80.4    0.207    0.999    0.170    0.052    0.053    0.000    0.000
RANGE:01.001-02.000    315      3.6    1.469    1.998    0.726    0.269    0.213    0.000    0.000
RANGE:02.001-03.000    140      1.6    2.502    2.993    1.006    0.356    0.200    0.000    0.000
RANGE:03.001-04.000     92      1.1    3.486    3.989    1.134    0.397    0.199    0.000    0.000
RANGE:04.001-05.000     55      0.6    4.417    4.928    1.380    0.499    0.367    0.000    0.000
RANGE:05.001-06.000     41      0.5    5.499    5.975    1.082    0.434    0.294    0.000    0.000
RANGE:06.001-07.000     40      0.5    6.401    6.847    1.295    0.377    0.092    0.000    0.000
RANGE:07.001-08.000     41      0.5    7.496    7.990    0.741    0.278    0.472    0.000    0.000
RANGE:08.001-09.000     12      0.1    8.545    8.950    0.700    0.419    0.074    0.000    0.000
RANGE:09.001-10.000     18      0.2    9.462    9.924    0.687    0.254    0.273    0.000    0.000
RANGE:10.001-20.000    217      2.5    15.80    19.95    0.445    0.232    0.055    0.000    0.000
RANGE:20.001-50.000     99      1.1    31.62    49.13    0.611    0.545    0.374    0.000    0.000
RANGE:50.001+          633      7.3     1182    46487    0.259    7.542    0.243    0.000    0.000
------------------- -------- -------- -------- -------- -------- -------- -------- -------- ------
SUMMARY               8684    100.0    87.43    46487    0.275    0.633    0.086    0.000    0.000
------------------- -------- -------- -------- -------- -------- -------- -------- -------- ------
```

Use the COST Command to Create Charge-Back Reports

To create charge-back reports, use the COST command with the TAB command.

For information about creating distributed cost reports, see the description of the COST command in the chapter “Commands.”

Example: COST Commands

The COST commands in the following example charge:

- One cent for every transaction
- 25 cents for every CPU second used
- 1/10 cent per file request
- 1/2 cent per file added

The TAB command uses these cost factors, creating a tabular report with a column showing the value of each variable for each transaction. The HDR1 and HDR2 operands assign those columns appropriate headings. The last variable, REPORT COST TOTAL, creates a column that shows the sum of the costs for each row in the report.

```
COST TRAN USE   .01
COST TRAN CPU   .25
TAB TRANID TRAN USE                                    AND,
           TRAN USE  COST HDR1(TRAN USE) HDR2(COST)    AND,
           TRAN CPU                                    AND,
           TRAN CPU  COST HDR1(TRAN CPU) HDR2(COST)    AND,
           REPORT COST TOTAL
END
RUN
```

A report like the following is generated:

```
  TRAN      TRAN   TRAN USE   TRAN   TRAN CPU   TOTAL
   ID        USE      COST     CPU      COST     COST
-------- -------- -------- -------- -------- --------
X25T          57     $0.57   0.132     $1.88    $2.45
CSNE          49     $0.49   0.010     $0.12    $0.61
TCP           46     $0.46   0.572     $6.57    $7.03
KCP           45     $0.45   7.581    $85.28   $85.73
PSPI          43     $0.43   0.097     $1.04    $1.47
POSS          37     $0.37   0.035     $0.32    $0.69
VGBC          13     $0.13   0.055     $0.18    $0.31
JJJ           10     $0.10   0.221     $0.55    $0.65
CSPQ          10     $0.10   0.113     $0.28    $0.38
CATD           9     $0.09   0.105     $0.24    $0.33
CSSN           8     $0.08   0.071     $0.14    $0.22
CSGM           8     $0.08   0.007     $0.01    $0.09
RAPL           4     $0.04   0.011     $0.01    $0.05
SPAM           3     $0.03   0.132     $0.10    $0.13
PINV           2     $0.02   1.486     $0.74    $0.76
PADI           2     $0.02   0.350     $0.17    $0.19
RAPP           2     $0.02   0.098     $0.05    $0.07
-------- -------- -------- -------- -------- --------
SUMMARY      348     $3.47  11.022   $ 97.68   $101.16
-------- -------- -------- -------- -------- - ------
```

Create Charge-Back Reports Organized by Group

Use the COST command with the GROUP command to create charge-back reports organized by groups.

Example: COST Command used with GROUP Command

The following commands use terminal IDs to define company divisions and list their corresponding charges in the report:

```
GROUP TERMID CO.NORTHERN.DIVISION(L4*)
GROUP TERMID CO.SOUTHERN.DIVISION(L3*)
GROUP TERMID CO.EASTERN.DIVISION(L2*)
GROUP TERMID CO.WESTERN.DIVISION(L1*)
COST  TRAN USE  .01
COST  TRAN CPU  .25
TAB TERMID(CO*) HDR1(DEPARTMENT)  HDR2(NAME),
    TRAN USE                                AND,
    TRAN USE COST               HDR2(COST)  AND,
    TRAN CPU COST HDR1(CPU)     HDR2(COST)  AND,
    REPORT COST TOTAL
END
RUN
```

A report like the following is generated:

```
      DEPARTMENT          TRAN      TRAN      CPU      TOTAL
        NAME               USE      COST      COST      COST
--------------------  --------  --------  --------  -------
CO.WESTERN.DIVISION     4415     $44.15    $32.29    $76.44
CO.NORTHERN.DIVISION    2224     $22.24    $15.15    $37.39
CO.EASTERN.DIVISION     2035     $20.35    $15.79    $36.14
CO.SOUTHERN.DIVISION    2024     $20.24    $43.73    $63.97
--------------------  --------  --------  --------  --------
SUMMARY                10698    $106.98   $106.97   $213.94
--------------------  --------  --------  --------  --------
```

TAB2 Command

Use the TAB2 command to present more information about each resource reported upon by the TAB command. You cannot use the TAB2 command without the TAB command.

Limit TAB2 Report Use the FOR Operand

The FOR operand with the TAB2 command limits the number of identifiers included in a report for each identifier listed by the TAB command.

Example 1: TAB and TAB2 Commands

The following TAB and TAB2 commands create a report showing terminals and the five transactions used the most by each:

```
TAB  TERMID       TRAN USE   AND,
                  TRAN LIFE  AND,
                  TRAN CPU
TAB2 TRANID FOR 5 TRAN USE   AND,
                  TRAN LIFE  AND,
                  TRAN CPU
END
RUN
```

A report like the following, shown in part, is generated:

```
TERM     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN
ID       USE      LIFE     CPU      ID       USE      LIFE     CPU
-------- -------- -------- -------- -------- -------- -------- --------
WA19          65    1.943    0.037  CSPG          40    0.033    0.002
                                    GLAE          23    5.193    0.098
                                    CSSN           1    5.527    0.080
                                    CSGM           1    0.037    0.002
                                    -------- -------- -------- --------
                                    SUMMARY       65    1.943    0.037
                                    -------- -------- -------- --------
4A93          53    18.37    0.116  WISB          51    19.01    0.121
                                    CSSN           1    4.460    0.019
                                    CSGM           1    0.012    0.002
                                    -------- -------- -------- --------
                                    SUMMARY       53    18.37    0.116
                                    -------- -------- -------- --------
BP3P          42    2.921    0.088  RAPP          42    2.921    0.088
0A0B          33    2.424    0.020  CSPG          18    0.022    0.002
                                    PADI          14    5.683    0.044
                                    CSGM           1    0.017    0.002
                                    -------- -------- -------- --------
                                    SUMMARY       33    2.424    0.020
                                    -------- -------- -------- --------
0A14          31    5.125    0.086  WISI          11    14.26    0.199
                                    CHAI          11    0.071    0.022
                                    PART           7    0.173    0.035
                                    CSGM           2    0.015    0.002
                                    -------- -------- -------- --------
                                    SUMMARY       31    5.125    0.086
                                    -------- -------- -------- --------
-------- -------- -------- -------- -------- -------- -------- --------
SUMMARY     8684    87.43    0.633             8684    87.43    0.633
-------- -------- -------- -------- -------- -------- -------- --------
-------- -------- -------- -------- -------- -------- -------- --------
```

Example 2: FOR Operand with TAB and TAB2 Commands

The FOR operand combined with both the TAB and TAB2 commands creates a table of transactions and the terminals associated with them:

```
TAB  TRANID FOR 3 TRAN USE   AND,
                  TRAN LIFE  AND,
                  TRAN CPU
TAB2 TERMID FOR 9 TRAN USE   AND,
                  TRAN LIFE  AND,
                  TRAN CPU
END
RUN
```

A report like the following is generated:

```
   TRAN      TRAN      TRAN      TRAN      TERM      TRAN      TRAN      TRAN
    ID        USE      LIFE       CPU       ID        USE      LIFE       CPU
-------- --------- --------- --------- --------- --------- --------- ---------
FXFR        2482    0.361     0.072    F014        2482    0.361     0.072
PSPI         939    0.248     0.092    8A21         197    0.219     0.084
.                                      F006         116    0.327     0.158
.                                      0A09          67    0.198     0.103
.                                      0A02          64    0.224     0.103
.                                      8A03          60    0.228     0.060
.                                      8A09          57    0.163     0.048
.                                      F012          41    0.271     0.106
.                                      4A64          41    0.235     0.079
.                                      8A24          40    0.370     0.102
.                                      --------- --------- --------- ---------
.                                      SUMMARY      683     1305     517.8
.                                      --------- --------- --------- ---------
CSNE         699    0.026     0.007    N/A          699    0.026     0.007
-------- --------- --------- --------- --------- --------- --------- ---------
SUMMARY     4120    0.279     0.066               8684    87.43     0.633
-------- --------- --------- --------- --------- --------- --------- ---------
-------- --------- --------- --------- --------- --------- --------- ---------
```

Limit TAB2 Reports Using the TOP or BOTTOM Operand

The TOP and BOTTOM operands, like the FOR operand, limit reports organized by identifier to the number of identifiers specified.

However, the TOP and BOTTOM operands save storage because:

- The FOR operand tests identifiers only after all the data has been read and processed.
- The TOP and BOTTOM operands cause the Report Writer to test each record as it is read and discard records that do not qualify.

The TOP and BOTTOM operands are appropriate for ID options with many unique identifiers, such as TRANNUM.

The TAB command in the following example creates a report on the terminal response time of transactions. The TAB2 command shows the four transactions that had the longest lifetimes for each terminal. Processing similar commands using the FOR operand instead of the TOP operand would require much more storage.

```
TAB  TERMID TERM RESP
TAB2 TRANNUM TRAN LIFE TOP 4
END
RUN
```

A report like the following, shown in part, is generated:

```
TERM       TERM     TRAN     TRAN
ID         RESP    NUMBER    LIFE
-------- -------- -------- --------
CNSL      2.149   0000020   13.30
.                 0000019   0.016
.                 -------- --------
.                 SUMMARY   6.663
.                 -------- --------
ZA00      0.278   0000054   306.6
.                 0000058   0.647
.                 0000059   0.486
.                 0000053   0.264
.                 -------- --------
.                 SUMMARY   77.01
.                 -------- --------
0A19      0.202   0000297    2390
.                 0000613    2283
.                 0000427    2209
.                 0000229    1600
.                 -------- --------
.                 SUMMARY    2121
.                 -------- --------
  .         .        .         .
-------- -------- -------- --------
SUMMARY   0.275             4176
-------- -------- -------- --------
-------- -------- -------- --------
```

Chapter 9: Plot Reports

This section contains the following topics:

What You Need to Know (see page 225)
PLOT Command—Create a Simple PLOT Report (see page 226)
PLOT2 Command—Plot Response Time (see page 231)
VPLOT Command—Create a Simple Report (see page 234)
VPLOT2 Command—Create a Simple VPLOT2 Report (see page 238)
MPLOT Command—Used with One and Multiple Variables (see page 239)
HPLOT Command (see page 241)

What You Need to Know

This chapter describes and gives examples of how to generate plot reports using the PLOT, PLOT2, VPLOT, VPLOT2, MPLOT, and HPLOT commands.

Plot reports plot the value of a variable for a group of resources either against time or against another variable. Plot reports fall into three categories:

- Horizontal plot reports, generated by the PLOT and PLOT2 commands, plot a variable horizontally across time.
- Vertical plot reports, generated by the VPLOT, VPLOT2, and MPLOT commands, plot a variable vertically across time.
- Distribution plot reports plot one variable on the vertical axis and another on the horizontal axis. The HPLOT command used with either the VPLOT or the MPLOT command generate these plots.

Understand the following:

- Report Writer commands
- Descriptions of Report Writer variables and lists of the ID options you can use with each class of variable
- Descriptions of ID options and their use with commands and variables

More information:

Commands (see page 39)
ID Options (see page 167)
Variables (see page 109)

PLOT Command—Create a Simple PLOT Report

To create a simple plot report, use the PLOT command followed by a variable to plot.

The following commands produce a plot of average terminal response time, with each detail line representing an hour, for the morning of a single day:

```
PLOT TERM RESPONSETIME
  FROM 12/07/08 01:00:00
  TO   12/07/08 09:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:

```
SHIFT NONE
EACH   HOUR                                  TERMINAL AVERAGE RESPONSE TIME
PERIOD NONE
                  0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X    -
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 01.00.00 .-------------------------------------------------------------       .         .         .         .     0.206
         02.00.00 .--------------------------------       .         .         .         .         .         .         .     0.106
         03.00.00 .-----------------------------------.         .         .         .         .         .         .     0.128
         04.00.00 .----------------------------------   .         .         .         .         .         .         .     0.120
         05.00.00 .----------------------------------   .         .         .         .         .         .         .     0.120
         06.00.00 .-----------------------------------------------------------------------------------------------------     0.332
         07.00.00 .-------------------------------------------------------------------------       .         .         .     0.242
         08.00.00 .-----------------------------------------------------------------------------       .         .         .     0.240
SUMMARY:          .         .         .         .         .         .         .         .         .         .         . SUMMARY:
         RANGE:   .-----------------------------------------------------   .         .         .         .         .     0.186
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                  0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X     -
SYMBOLS:  “X” =  TERMINAL AVERAGE RESPONSE TIME        “-” =  GLOBAL TERMINAL AVERAGE RESPONSE TIME
```

Plot the Variable Value for a Specified Resource

You can plot the performance of one or more individual resources (such as transactions, terminals, or operators). Follow the variables to plot with an ID option and a list of one or more identifiers (such as terminal ID or transaction ID). You can use generic characters to specify a group of identifiers.

Example: PLOT Command with Variable and ID option

The following PLOT command includes the variable TERM RESP, followed by the ID option TERMID and the identifier 0A. The average response time of terminals whose IDs begin with the characters *0A* plots:

```
PLOT TERM RESP TERMID(0A*)
  FROM 12/07/08 01:00:00
  TO   12/07/08 09:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   HOUR                                   TERMINAL AVERAGE RESPONSE TIME
PERIOD NONE
                  0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X     -
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 01.00.00 .XXXXXXXXXXX----------------------------------------------       .         .         .         . 0.040 0.206
         02.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXX-----       .         .         .         .         .         .         . 0.090 0.106
         03.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX-.         .         .         .         .         .         . 0.125 0.128
         04.00.00 .XXXXXXXXXXXXXXXXXXXXXX-----------   .         .         .         .         .         .         . 0.083 0.120
         05.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXX-----   .         .         .         .         .         .         . 0.104 0.120
         06.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 0.332 0.332
         07.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX---------------      .         .         . 0.194 0.242
         08.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX       .         .         . 0.240 0.240
SUMMARY:          .         .         .         .         .         .         .         .         .         .         . SUMMARY:
         RANGE:   .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX-----------   .         .         .         .         . 0.151 0.186
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                  0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X     -
SYMBOLS:  "X" =  TERMINAL AVERAGE RESPONSE TIME        "-" =  GLOBAL TERMINAL AVERAGE RESPONSE TIME
```

The average response time of 0A terminals is plotted with *X*s. Also the average response time for all terminals is also plotted with hyphens. You can easily compare the two.

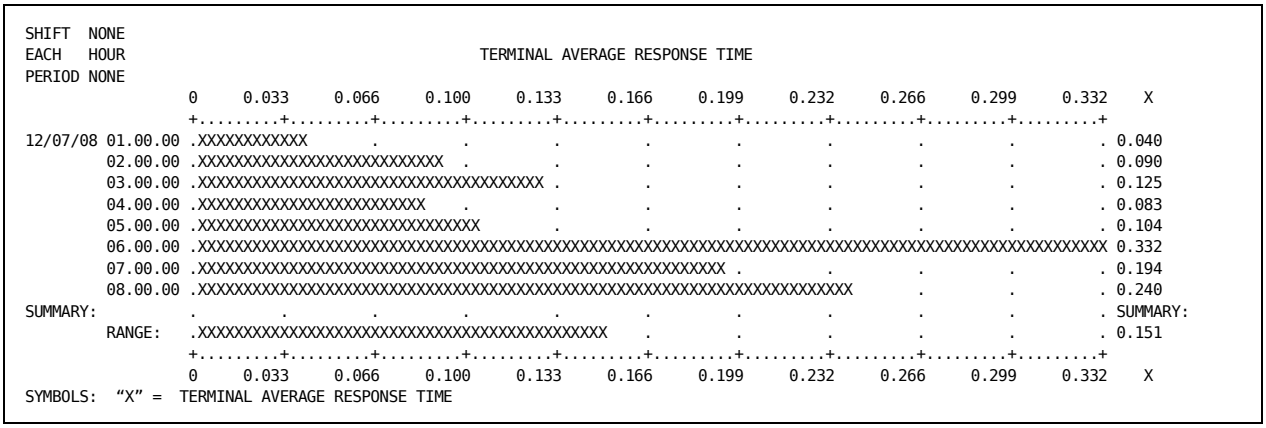
ALONE Operand—Exclude Global Values from a Plot

Use the ALONE operand to plot the activity of one or more identifiers without including a plot of global activity in the report.

Suppose you want to plot the response time for 0A terminals, but you do not want to plot the average response time for all terminals. You can use the same commands as in the previous example but add the ALONE operand, as follows:

```
PLOT TERM RESP TERMID(0A*) ALONE
  FROM 12/07/08 01:00:00
  TO   12/07/08 09:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:



```
SHIFT  NONE
EACH   HOUR                                    TERMINAL AVERAGE RESPONSE TIME
PERIOD NONE
                 0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X
                 +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 01.00.00 .XXXXXXXXXXX       .         .         .         .         .         .         .         .         . 0.040
         02.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXX  .         .         .         .         .         .         .         . 0.090
         03.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX .         .         .         .         .         .         . 0.125
         04.00.00 .XXXXXXXXXXXXXXXXXXXXXXX    .         .         .         .         .         .         .         . 0.083
         05.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXX        .         .         .         .         .         .         . 0.104
         06.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 0.332
         07.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX .         .         .         .         . 0.194
         08.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX       .         .         . 0.240
SUMMARY:          .         .         .         .         .         .         .         .         .         .         . SUMMARY:
         RANGE:   .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX    .         .         .         .         .         . 0.151
                 +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                 0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X
SYMBOLS:  "X" =  TERMINAL AVERAGE RESPONSE TIME
```

SCALE Operand—Change the Scale of a PLOT Report

By default, the highest value of a plot scale equals the highest value reached by a variable in the plot. For variables representing percentages, 100 percent. Use the SCALE operand to set your own scale.

If the variable you are plotting yields integer values, you can specify only an integer value with the SCALE operand. If the variable yields real values, the scale you specify must include a decimal point.

Example: PLOT Command with SCALE Operand

To change the maximum value of the scale used in the preceding example to 0.500 seconds, add the SCALE(0.500) operand, as follows:

```
PLOT TERM RESP TERMID(0A*) ALONE SCALE(0.500)
  FROM 12/07/08 01:00:00
  TO   12/07/08 09:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   HOUR                                    TERMINAL AVERAGE RESPONSE TIME
PERIOD NONE
                  0     0.050     0.100     0.150     0.200     0.250     0.300     0.350     0.400     0.450     0.500    X
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 01.00.00 .XXXXXXX .         .         .         .         .         .         .         .         .         . 0.040
         02.00.00 .XXXXXXXXXXXXXXXX .         .         .         .         .         .         .         .         . 0.090
         03.00.00 .XXXXXXXXXXXXXXXXXXXXXXXX    .         .         .         .         .         .         .         . 0.125
         04.00.00 .XXXXXXXXXXXXXXX  .         .         .         .         .         .         .         .         . 0.083
         05.00.00 .XXXXXXXXXXXXXXXXXXX        .         .         .         .         .         .         .         . 0.104
         06.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   .         .         .         . 0.332
         07.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.         .         .         .         .         .         . 0.194
         08.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX .         .         .         .         .         . 0.240
SUMMARY:          .         .         .         .         .         .         .         .         .         .         . SUMMARY:
         RANGE:   .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX         .         .         .         .         .         .         . 0.151
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                  0     0.050     0.100     0.150     0.200     0.250     0.300     0.350     0.400     0.450     0.500    X
SYMBOLS:  "X" =  TERMINAL AVERAGE RESPONSE TIME
```

CHARn Operands—Specify Plot Characters

Use the CHAR1 operand to specify the character used to plot the resource specified by an ID option. Use the CHAR2 operand to specify the character used to plot global activity.

Example: CHAR*n* Operand

The following commands plot the usage of transaction PSPI using the character *P* and the usage of all transactions using plus signs (+):

```
PLOT TRAN USE TRANID(PSPI) CHAR1(P) CHAR2(+)
  FROM 12/07/08 01:00:00
  TO   12/07/08 09:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   HOUR                                        TRANSACTION COUNT
PERIOD NONE
                  0     22.80     45.60     68.40     91.20     114.0     136.8     159.6     182.4     205.2     228.0    P    +
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 01.00.00 .PPPP++++++++++++++++++++++++++++++++++++++++++++++++++   .         .         .         .         .         .     9  105
         02.00.00 .PPPPPPPPPPPP++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++    .         .         .         .         .    29  149
         03.00.00 .PPPP++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++   .         .         .         .         .     9  151
         04.00.00 .PPPPPPP++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++    .         .         .         .         .    17  194
         05.00.00 .PPPPPPPPPPP
```

RATE Option—Convert Counts to Rates

Use the RATE option to plot a rate (number per second) rather than a count.

Example: Plot a Rate

The following commands plot the usage rate for transaction PSPI against the total transaction rate:

```
PLOT TRAN USE RATE TRANID(PSPI) CHAR1(P) CHAR2(+)
  FROM 12/07/08 01:00:00
  TO   12/07/08 09:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:

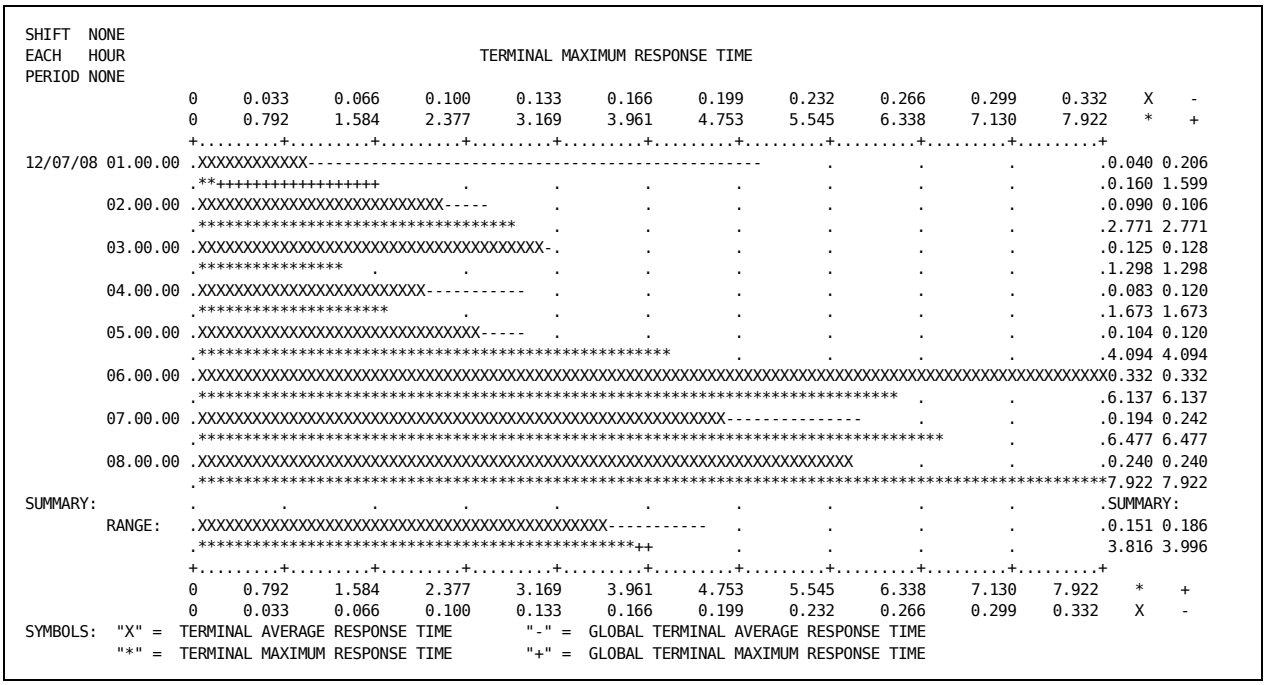
```
SHIFT  NONE
EACH   HOUR                                          TRANSACTION COUNT
PERIOD NONE
                 0      0.01      0.01      0.02      0.02      0.03      0.04      0.04      0.05      0.05      0.06    P    +
                 +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 01.00.00 .++++++++++++++++++++++++++++++++++++++++++++++++++++++++++          .         .         .         .         .  0.00 0.03
         02.00.00 .PPPPPPPPPPPPPPPPP++++++++++++++++++++++++++++++++++++++++++++++++++  .         .         .         .         .  0.01 0.04
         03.00.00 .++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  .         .         .         .         .  0.00 0.04
         04.00.00 .++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++      .         .  0.00 0.05
         05.00.00 .PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  0.02 0.06
         06.00.00 .++++++++++++++++++++++++++++++++++++++++++++++++++++          .         .         .         .         .  0.00 0.03
         07.00.00 .PPPPPPPPPPPPPPPPP++++++++++++++++++++++++++++++++++          .         .         .         .         .  0.01 0.03
         08.00.00 .++++++++++++++++++++++++++++++++++      .         .         .         .         .         .         .  0.00 0.02
SUMMARY:          .         .         .         .         .         .         .         .         .         .         . SUMMARY:
         RANGE:   .++++++++++++++++++++++++++++++++++++++++++++++++++++          .         .         .         .         .  0.00 0.03
                 +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                 0      0.01      0.01      0.02      0.02      0.03      0.04      0.04      0.05      0.05      0.06    P    +

SYMBOLS:  "P" =  TRANSACTION COUNT                    "+" =  GLOBAL TRANSACTION COUNT
```

PLOT2 Command—Plot Response Time

The following commands generate a report that compares the response time of 0A terminals with the maximum terminal response time for 0A terminals. The average response time and average maximum response time for all terminals is also plotted.

```
PLOT  TERM RESP    TERMID(0A++)
PLOT2 TERM MAXRESP TERMID(0A++)
  FROM 12/07/08 01:00:00
  TO   12/07/08 09:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:



```
SHIFT  NONE
EACH   HOUR                                    TERMINAL MAXIMUM RESPONSE TIME
PERIOD NONE
                  0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X    -
                  0     0.792     1.584     2.377     3.169     3.961     4.753     5.545     6.338     7.130     7.922    *    +
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 01.00.00 .XXXXXXXXXXX----------------------------------------------------       .         .         .         .0.040 0.206
                  .**+++++++++++++++++         .         .         .         .         .         .         .         .0.160 1.599
         02.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXX-----       .         .         .         .         .         .         .0.090 0.106
                  .**********************************    .         .         .         .         .         .         .2.771 2.771
         03.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX-.         .         .         .         .         .         .0.125 0.128
                  .****************   .         .         .         .         .         .         .         .1.298 1.298
         04.00.00 .XXXXXXXXXXXXXXXXXXXXXXXX-----------   .         .         .         .         .         .         .0.083 0.120
                  .*********************        .         .         .         .         .         .         .         .1.673 1.673
         05.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXX-----   .         .         .         .         .         .         .0.104 0.120
                  .************************************************       .         .         .         .         .4.094 4.094
         06.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0.332 0.332
                  .*************************************************************************** .         .         .6.137 6.137
         07.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX--------------      .         .         .0.194 0.242
                  .*************************************************************************************       .         .6.477 6.477
         08.00.00 .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX       .         .         .0.240 0.240
                  .*************************************************************************************************************7.922 7.922
SUMMARY:          .         .         .         .         .         .         .         .         .         .         .SUMMARY:
         RANGE:   .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX-----------   .         .         .         .         .0.151 0.186
                  .************************************************++         .         .         .         .          3.816 3.996
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                  0     0.792     1.584     2.377     3.169     3.961     4.753     5.545     6.338     7.130     7.922    *    +
                  0     0.033     0.066     0.100     0.133     0.166     0.199     0.232     0.266     0.299     0.332    X    -
SYMBOLS:  "X" =  TERMINAL AVERAGE RESPONSE TIME        "-" =  GLOBAL TERMINAL AVERAGE RESPONSE TIME
          "*" =  TERMINAL MAXIMUM RESPONSE TIME        "+" =  GLOBAL TERMINAL MAXIMUM RESPONSE TIME
```

PLOT2 RATE Command—Convert Counts to Rates

Use the RATE option to plot a rate (number per second) rather than a count.

Example: RATE Option on the PLOT2 Command

In the following example, the PLOT command plots the use of PSPI against the total system transaction use. The PLOT2 command plots the transaction rate for PSPI against the total system transaction rate:

```
PLOT  TRAN USE      TRANID(PSPI) CHAR1(T) CHAR2(+)
PLOT2 TRAN USE RATE TRANID(PSPI) CHAR1(R) CHAR2(-)
  FROM 12/07/08 02:00:00
  TO   12/07/08 08:00:00
  EACH HOUR
END
RUN
```

A report like the following is generated:

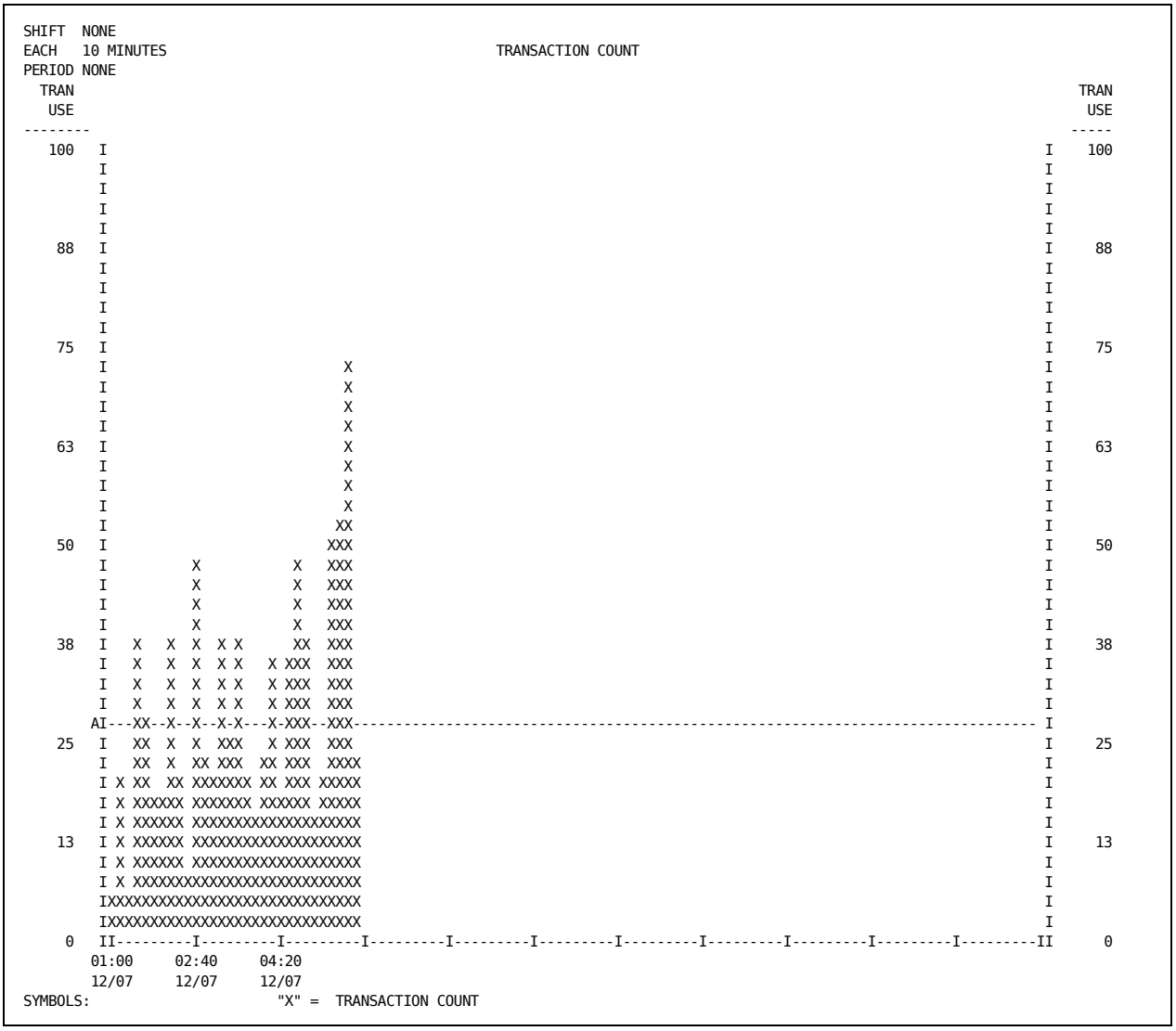
```
SHIFT  NONE
EACH   HOUR                                          TRANSACTION COUNT
PERIOD NONE
                  0     22.80     45.60     68.40     91.20     114.0     136.8     159.6     182.4     205.2     228.0   T     +
                  0     0.01      0.01      0.02      0.02      0.03      0.04      0.04      0.05      0.05      0.06    R     -
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
12/07/08 02.00.00 .TTTTTTTTTTTT++++++
```

VPLOT Command—Create a Simple Report

The following commands plot terminal response time between 1:00 and 6:00 a.m.:

```
VPLOT TRAN USE
  FROM 12/07/08 01:00:00
  TO   12/07/08 06:00:00
EACH 10 MINUTES
END
RUN
```

A report like the following is generated:

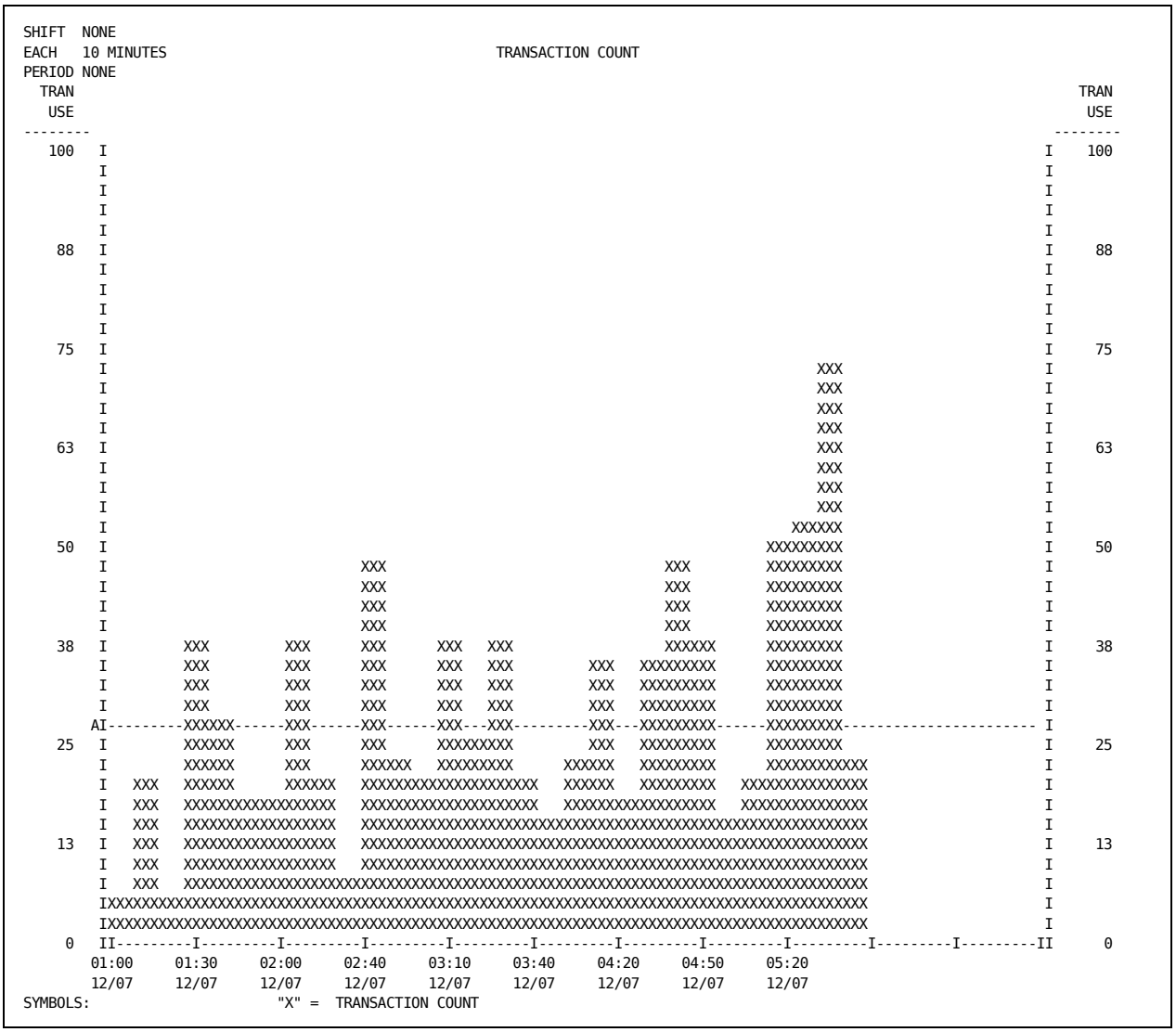


SCALE(3X) Operands—Widen the Scale of a VPLOT Report

The following commands resemble the previous example except for the addition of the SCALE(3X) operand, which widens the resulting plot:

```
VPLOT TRAN USE SCALE(3X)
  FROM 12/07/08 01:00:00
  TO   12/07/08 06:00:00
EACH 10 MINUTES
END
RUN
```

A report like the following is generated:

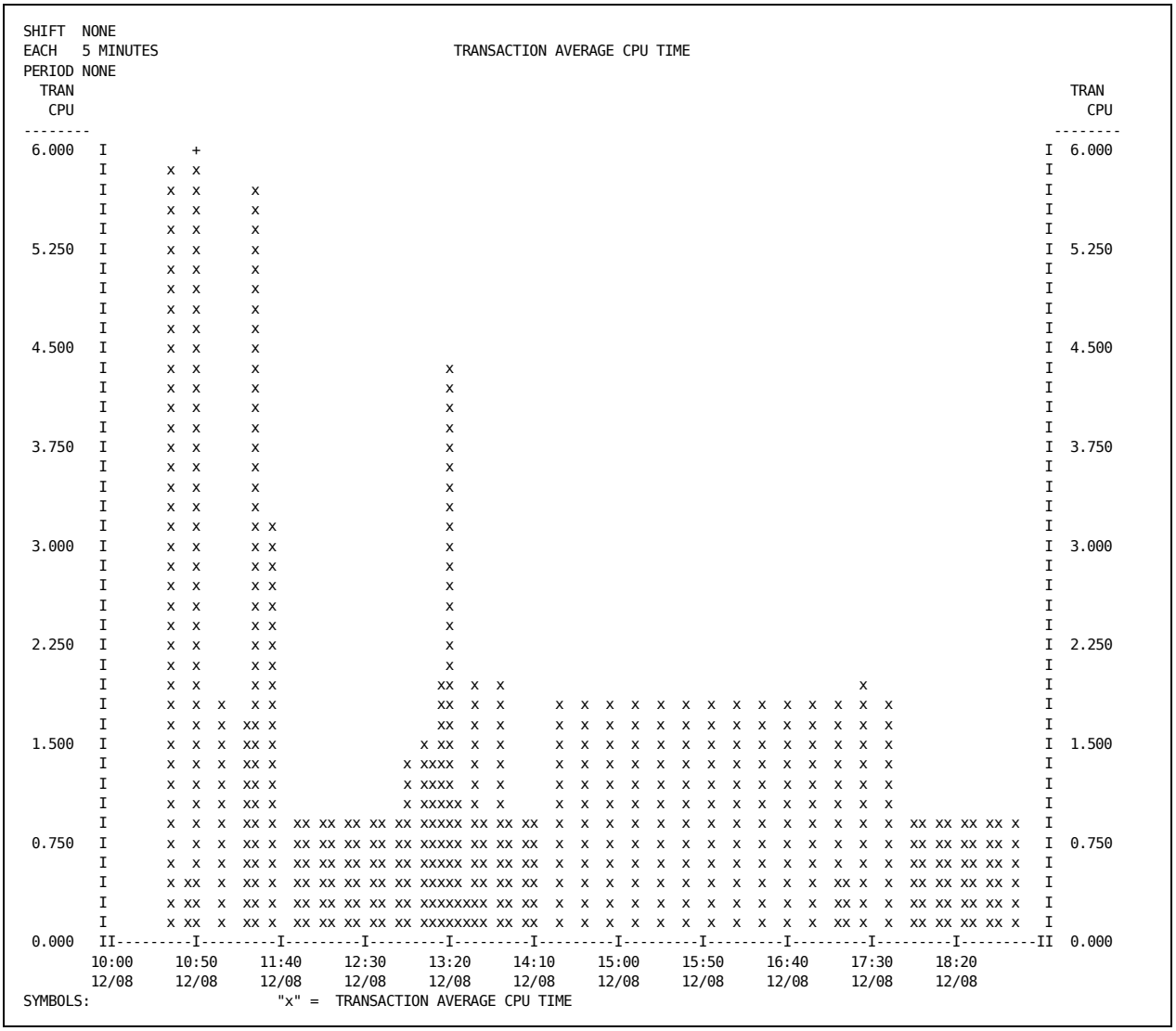


SCALE (n) Operand—Specify the Maximum Scale Value

The following example uses the SCALE(6.000) operand to plot transaction CPU use with a maximum scale value of 6.000 seconds. The CHAR1 operand specifies *x* as the plot character.

```
VPLOT TRAN CPU TOTAL CHAR1(x) SCALE(6.000)
  FROM 12/08/08 10:00:00
  TO   12/08/08 19:00:00
  EACH 5 MINUTES
END
RUN
```

A report like the following is generated:



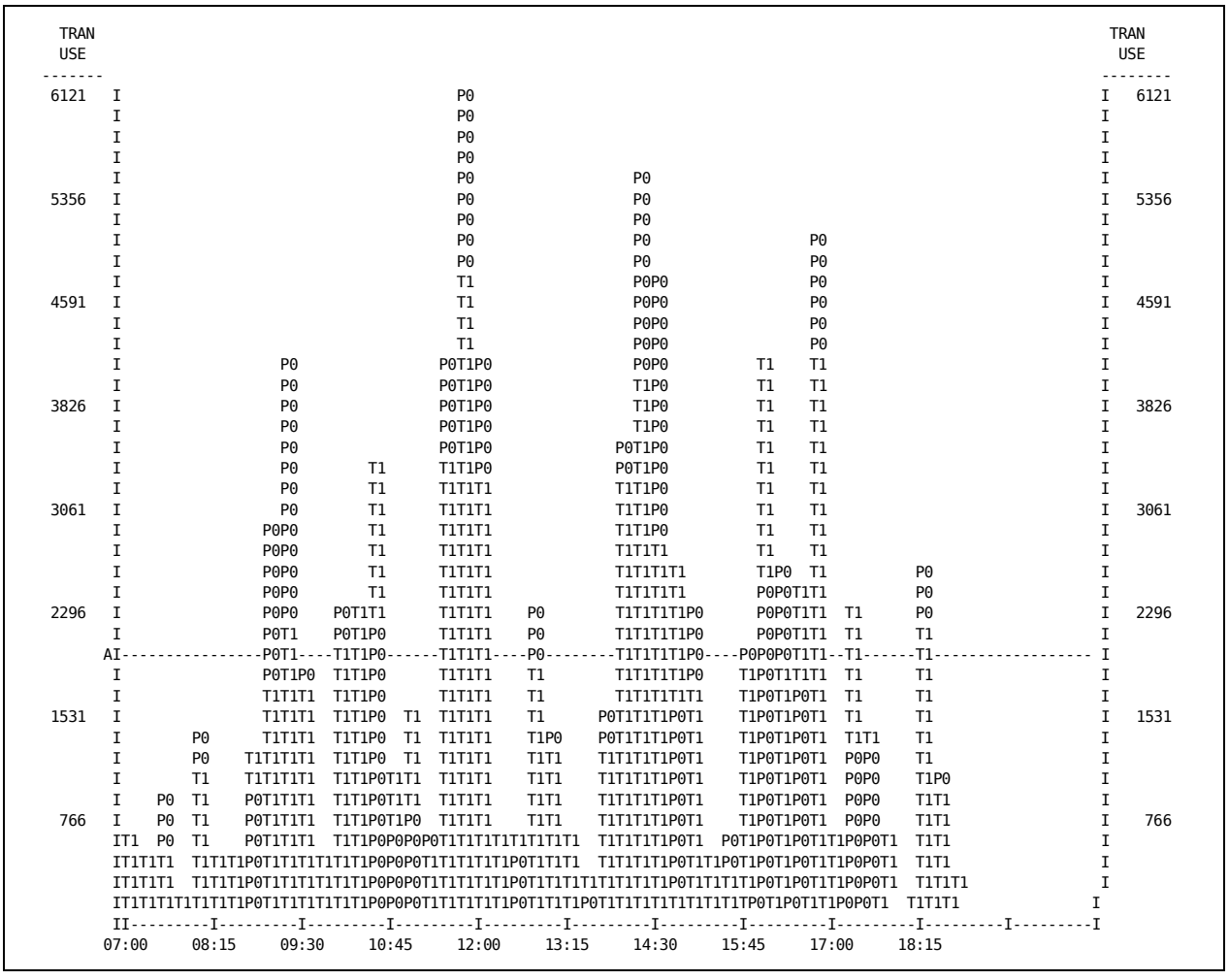
STACK=id Operand—Replace the Plot Character

Use the STACK=*id* operand with the VPLOT command to replace the plot character with the identifiers being plotted. Specify from most active to least active starting at the bottom of the plot. You can specify the *m*X option of the SCALE operand to print more characters of each identifier.

The following commands create a vertical plot of transaction use. The STACK=CICSID operand specifies to plot CICS IDs. The SCALE(2X) operand widens each plot column to two characters so the entire CICS ID can be printed.

```
VPLOT TRAN USE SCALE(2X) STACK=CICSID
END
RUN
```

A report like the following is generated:



```
        03/14     03/14     03/14     03/14     03/14     03/14     03/14     03/14     03/14     03/14
```

VPLOT2 Command—Create a Simple VPLOT2 Report

The following example uses the VPLOT and VPLOT2 commands to plot both transaction use and terminal response time:

```
VPLOT  TRAN USE  SCALE(50)    CHAR1(X)
 VPLOT2 TERM RESP SCALE(1.000) CHAR1(O)
  EACH 5 MINUTES
  FROM 12/07/08 00:00:00
  TO   12/07/08 09:00:00
END
RUN
```

SCALE(COMMON) Operand—Use the Same Scale

The following commands plot transaction use and data set use. The SCALE(COMMON) operand uses the same scale for both variables being plotted.

```
VPLOT  TRAN USE    CHAR1(X)
 VPLOT2 DATASET USE CHAR1(O) SCALE(COMMON)
  FROM 12/07/08 00:00:00
  TO   12/07/08 09:00:00
  EACH 10 MINUTES
END
RUN
```

MPLOT Command—Used with One and Multiple Variables

Use the MPLOT command to produce a vertical plot similar to the one produced by the VPLOT command. The difference is that MPLOT prints only the outline of the columns.

MPLOT allows you to request multiple variables on the same report.

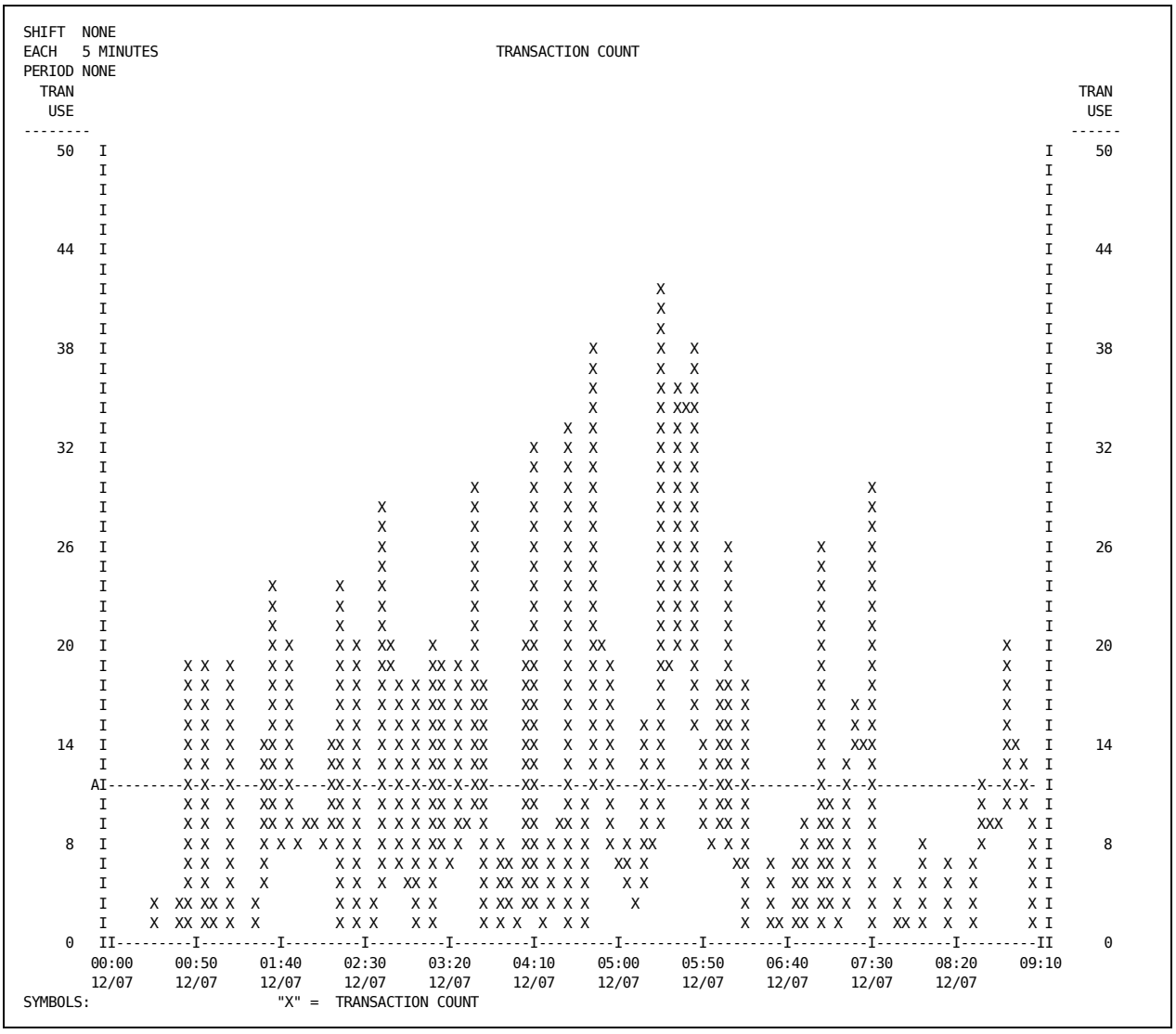
Example: Using Variables with the MPLOT Command

- Using One Variable with the MPLOT Command

  The following example uses the MPLOT command to plot terminal response times:

```
MPLOT TRAN USE SCALE(50)
  EACH 5 MINUTES
  FROM 12/07/08 00:00:00
  TO   12/07/08 23:59:59
END
RUN
```

A report like the following is generated:



- Using Two Variables with the MPLOT Command

  The following example uses the MPLOT command to plot transaction use with Xs and terminal response time with Os:

```
MPLOT TRAN USE  SCALE(50)    CHAR1(X) AND,
      TERM RESP SCALE(1.000) CHAR1(O)
  EACH 5 MINUTES
  FROM 12/07/08 00:00:00
  TO   12/07/08 23:59:59
END
RUN
```

- Using Three Variables with the MPLOT Command

  The following commands add of a third variable, DATASET USE, which is plotted with plus signs (+):

```
FROM 12/07/08 00:00:00
TO   12/07/08 09:00:00
EACH 5 MINUTES
MPLOT TRAN USE    SCALE(50)    CHAR1(X) AND,
      TERM RESP   SCALE(1.000) CHAR1(O) AND,
      DATASET USE SCALE(75)    CHAR1(+)
END
RUN
```

HPLOT Command

Use the HPLOT command to plot the distribution of one variable based on the value of another variable. The variable you specify with the HPLOT command is plotted along the horizontal axis. The second variable, specified with either the VPLOT or the MPLOT command, is plotted vertically.

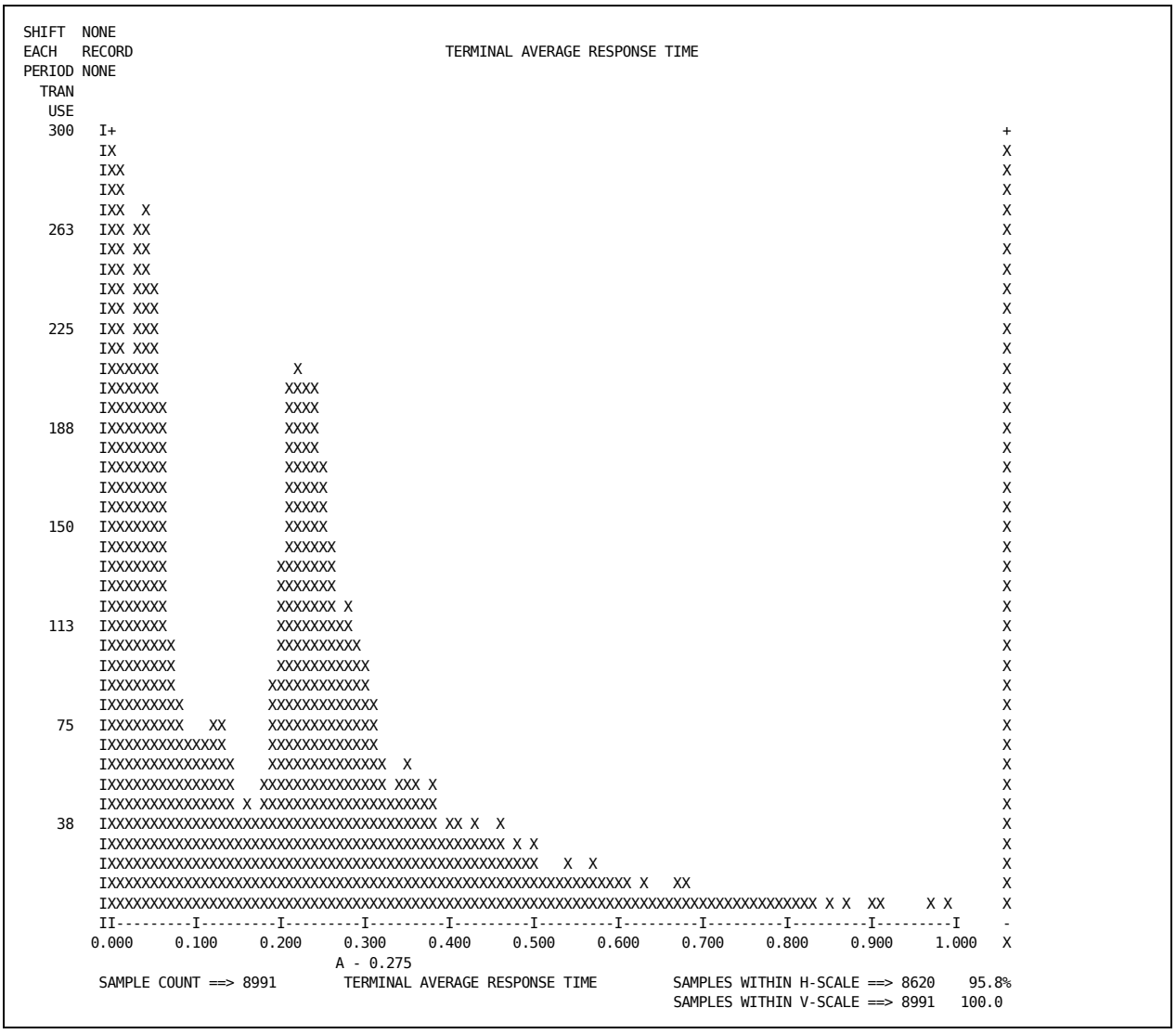
Use the HPLOT command with either the MPLOT or VPLOT command. The HPLOT command must follow the MPLOT or VPLOT command. To set the plot character for an HPLOT report, specify the CHAR1 operand with the MPLOT or VPLOT command.

SCALE Operand—Used with HPLOT and VPLOT

The following example uses the HPLOT and VPLOT commands to produce a terminal response time distribution plot:

```
VPLOT TRAN USE  SCALE(300)
 HPLOT TERM RESP SCALE(1.000)
 EACH RECORD
END
RUN
```

A report like the following is generated:



EACH RECORD Command—Used with VPLOT and HPLOT

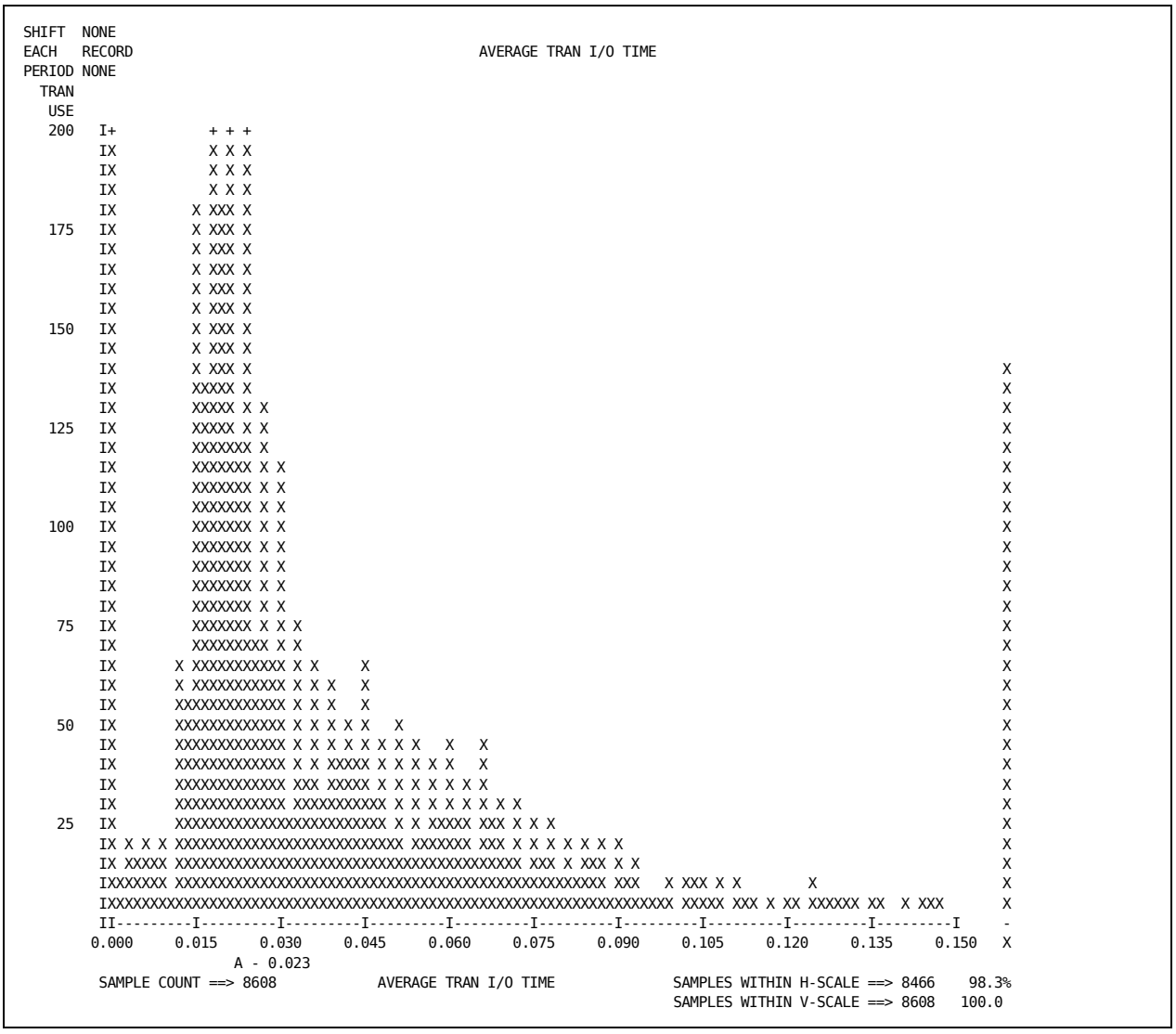
When you use the command EACH RECORD, always specify the SCALE operand with the HPLOT command.

The following commands are similar to the commands in the previous example with the following differences:

- The interval has been changed from each 5 minutes to each record
- The SCALE operand has been specified with both the VPLOT and HPLOT commands

```
VPLOT TRAN USE      SCALE(200)   CHAR1(X)
 HPLOT TRAN CPUTIME  SCALE(0.150)
 EACH RECORD
END
RUN
```

The previous commands generate a report like the following:

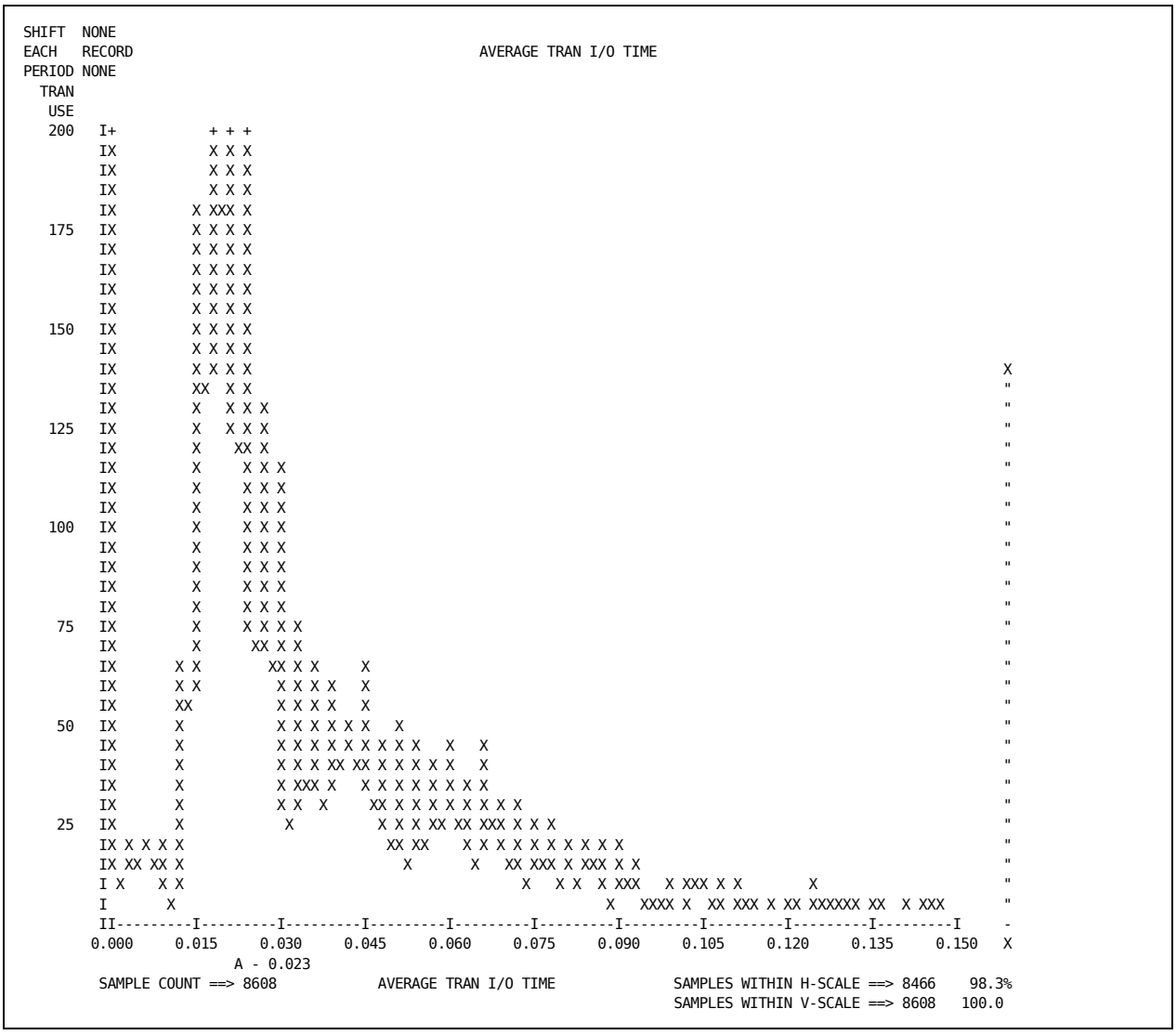


EACH RECORD Command—Used with MPLOT and HPLOT

The following commands are identical to the commands in the previous example except that MPLOT has been used instead of VPLOT.

```
MPLOT TRAN USE      SCALE(200)
HPLOT TRAN CPUTIME  SCALE(0.150)
 EACH RECORD
END
RUN
```

A report like the following is generated:



SCALE Command—Change the Scale

The following commands are identical to the commands in the previous example with the following differences:

- The scale specified with the MPLOT command has been changed to 300
- The letter *M* has been specified as the plot character

```
MPLOT TRAN USE       SCALE(300)    CHAR1(M)
 EACH RECORD
END
RUN
```

Chapter 10: Graph Reports

This section contains the following topics:

GRAF Command—Create a GRAF Report (see page 247)
GRAF2 Command Restrictions (see page 255)

GRAF Command—Create a GRAF Report

Graph reports are graphs used to compare related identifiers.

The GRAF command produces a single-line bar graph of the performance of each identifier associated with a variable. For example, using the GRAF command with a TRANSACTION variable produces a bar graph of the variable for each transaction.

The following commands produce a graph of transaction lifetime. The log file can contain data that covers many months. Use the FROM and TO commands to limit the report. For example, limit the report to transactions active between 10:00 a.m. and 6:00 p.m. on 12/08/10.

```
GRAF TRAN LIFE SCALE(1.598)
  FROM 12/08/10 10:00:00
  TO   12/08/10 18:00:00
RANGE TRAN LIFE 0-2.000 DEFERRED
END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY                                    TRANSACTION AVERAGE LIFETIME
PERIOD NONE
                0     0.160     0.320     0.479     0.639     0.799     0.959     1.119     1.278     1.438     1.598    X
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
CRSQ            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 1.598
CATD            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   .         . 1.376
CSSF            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX .         .         .         .         .         . 0.767
END             .XXXXXXXXXXXXXXXXXXXXXXXX    .         .         .         .         .         .         .         . 0.403
VGBR            .XXXXXXXXXXXXXXXXXXXXXX      .         .         .         .         .         .         .         . 0.371
CSNE            .XXXXXXXXXXXX      .         .         .         .         .         .         .         .         . 0.211
PSPI            .XXXXXXXXX.         .         .         .         .         .         .         .         .         . 0.149
POSS            .XXXXXX   .         .         .         .         .         .         .         .         .         . 0.092
CSAC            .XX       .         .         .         .         .         .         .         .         .         . 0.038
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                0     0.160     0.320     0.479     0.639     0.799     0.959     1.119     1.278     1.438     1.598    X
SYMBOLS:  "X" =  TRANSACTION AVERAGE LIFETIME
```

GRAF JOBNAME—Graph by an Alternate Resource

By default, the GRAF command produces a report on the identifiers associated with the specified variable. To graph the value of the variable for resources of another type, specify an ID option with the GRAF command.

Example: Report Jobname Usage

The following example uses the ID option JOBNAME to produce a report on transaction usage by CICS jobname rather than by transaction ID. Transaction ID is the default for TRANSACTION variables.

```
GRAF JOBNAME TRANSACTION USE
END
RUN
```

A report like the following is generated:

```
                0     191.9     383.8     575.7     767.6     959.5      1151      1343      1535      1727      1919    X
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
CICSA           .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  1919
CICSB           .XXXXXXXXXXX       .         .         .         .         .         .         .         .         .   236
CICSTEST        .XXXXXXXXXX        .         .         .         .         .         .         .         .         .   219
CICSX1          .XXXXXX   .        .         .         .         .         .         .         .         .         .   108
CICSX2          .         .        .         .         .         .         .         .         .         .         .     6
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                0     191.9     383.8     575.7     767.6     959.5      1151      1343      1535      1727      1919    X

SYMBOLS:  “X” =  TRANSACTION USE
```

TRANID ID Option—Specify the Resources to be Included in a Graph

Use an ID option and identifiers with the GRAF command to specify the resources to include in a graph report. You can specify individual resources or groups of resources.

Example: TRANID ID Option on the GRAF Command

Use the TRANID ID option and the generic identifier (C*) to include in the report only the transactions whose names start with *C*:

```
GRAF TRAN LIFE TRANID(C*) SCALE(1.598)
  FROM 12/08/10 10:00:00
  TO   12/08/10 18:00:00
RANGE TRAN LIFE 0-2.000 DEFERRED
END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY                                        TRANSACTION AVERAGE LIFETIME
PERIOD NONE
                0     0.160     0.320     0.479     0.639     0.799     0.959     1.119     1.278     1.438     1.598    X
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
CRSQ            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 1.598
CATD            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   .         . 1.376
CSSF            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX .         .         .         .         .         . 0.767
CSNE            .XXXXXXXXXXXX      .         .         .         .         .         .         .         .         . 0.211
CSAC            .XX       .         .         .         .         .         .         .         .         .         . 0.038
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                0     0.160     0.320     0.479     0.639     0.799     0.959     1.119     1.278     1.438     1.598    X
SYMBOLS:  “X” =  TRANSACTION AVERAGE LIFETIME
```

FOR Operand—Limit Lines in a Graph

Use the FOR *n* operand to produce a report on the *n* resources with the highest or lowest values for the variable being graphed.

Example: FOR Operand

The following commands graph average transaction lifetime by terminal. The FOR operand limits the report to the 20 terminals that have the longest transaction lifetimes. The CHAR1 operand sets the plot character to *T.* A continuation comma is used to continue the GRAF command to a second line.

```
GRAF TERMID TRAN LIFETIME,
  FOR 20 CHAR1(T)
 END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY                                    TRANSACTION AVERAGE LIFETIME
PERIOD NONE
                0     45.47     90.94     136.4     181.8     227.3     272.8     318.3     363.7     409.2     454.7    T
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
AF2P            .TTTT
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY                                    TRANSACTION AVERAGE LIFETIME
PERIOD NONE
                0     10.00     20.00     30.00     40.00     50.00     60.00     70.00     80.00     90.00     100.0    T
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
AF2P            .TTTT
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY                                    TRANSACTION AVERAGE LIFETIME
PERIOD NONE
                  0     0.020     0.040     0.060     0.080     0.100     0.120     0.140     0.160     0.180     0.200    X
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
WARE              .XXXXXXXXX.         .         .         .         .         .         .         .         .         . 0.018
CSNE              .XXXXXXXXXXXX       .         .         .         .         .         .         .         .         . 0.026
CSPG              .XXXXXXXXXXXXXXX    .         .         .         .         .         .         .         .         . 0.032
CSPS              .XXXXXXXXXXXXXXXXXXXXXXXXXXX .         .         .         .         .         .         .         . 0.057
CSPQ              .XXXXXXXXXXXXXXXXXXXXXXXXXXXX          .         .         .         .         .         .         . 0.060
CSAC              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX       .         .         .         .         .         .         . 0.066
WISC              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX         .         .         .         .         .         . 0.080
POSS              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX         .         .         .         . 0.120
END               .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  .         .         .         . 0.135
PART              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX        .         . 0.162
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                  0     0.020     0.040     0.060     0.080     0.100     0.120     0.140     0.160     0.180     0.200    X

SYMBOLS:  "X" =  TRANSACTION AVERAGE LIFETIME
```

SHIFT Command—Create Reports on Specified Shifts

Use the SHIFT command to create reports that cover only certain hours of the day.

The following commands produce a report on:

- The five transactions active between 11:00 p.m. and 7:00 a.m. the week of January 5 2010
- The transactions contain names starting with *C* that have the shortest lifetimes

```
GRAF TRAN LIFETIME SCALE(1.0) TRANID(C*),
     FOR 5 SORT ASCENDING
  FROM 01/05/10 00.00.00
  TO   01/11/10 00.00.00
 SHIFT 23:00:00 07:00:00
END
RUN
```

A report like the following is generated:

```
                  0     0.100     0.200     0.300     0.400     0.500     0.600     0.700     0.800     0.900     1.000    X
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
CSTA              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX    .         .         .         .         .         .         . 0.354
CSSN              .XXXXXXXXXXXXXXXXXXXX
```

EACH Command—Create Reports with Multiple Graphs

By default, graph reports include the following:

- One detail line for each identifier
- Shows the value of the variable specified for that identifier during the time period covered in the report

Using the EACH command with the GRAF command produces a detail line for each identifier for each time period specified by the EACH command.

The following FROM, TO, and EACH commands produce a report on transaction lifetimes for a given shift each week for one month:

```
GRAF TRANSACTION USE
  FROM 01/05/10 00.00.00
  TO   02/01/10 00.00.00
  EACH WEEK
  SHIFT 23:00:00 07:00:00
END
RUN
```

A report like the following is generated:

```
                0     1.200     2.400     3.600     4.800     6.000     7.200     8.400     9.600     10.80     12.00    X
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
01/05/08 23.00.00 .         .         .         .         .         .         .         .         .         .
 CSTE             .XXXXXXXX .         .         .         .         .         .         .         .         .         .    1
CSTA              .XXXXXXXX .         .         .         .         .         .         .         .         .         .    1
                  .         .         .         .         .         .         .         .         .         .         .
 01/12/08 23.00.00 .         .         .         .         .         .         .         .         .         .         .
 CINQ             .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   12
CSSN              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX         .         .         .         .         .         .         .    5
OSSN              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX       .         .         .         .         .         .         .         .    4
PINQ              .XXXXXXXX .         .         .         .         .         .         .         .         .         .    1
                  .         .         .         .         .         .         .         .         .         .         .
 01/20/08 23.00.00 .         .         .         .         .         .         .         .         .         .         .
 JCLR             .XXXXXXXX .         .         .         .         .         .         .         .         .         .    1
EDII              .XXXXXXXX .         .         .         .         .         .         .         .         .         .    1
                  .         .         .         .         .         .         .         .         .         .         .
 01/27/08 23.00.00 .         .         .         .         .         .         .         .         .         .         .
 CINQ             .XXXXXXXXXXXXXXXXXXXXXXXXX    .         .         .         .         .         .         .         .    3
CUST              .XXXXXXXXXXXXXXXXX  .         .         .         .         .         .         .         .         .    2
CSSN              .XXXXXXXX .         .         .         .         .         .         .         .         .         .    1
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                  0     1.200     2.400     3.600     4.800     6.000     7.200     8.400     9.600     10.80     12.00    X

SYMBOLS:  “X” =  TRANSACTION USE.
```

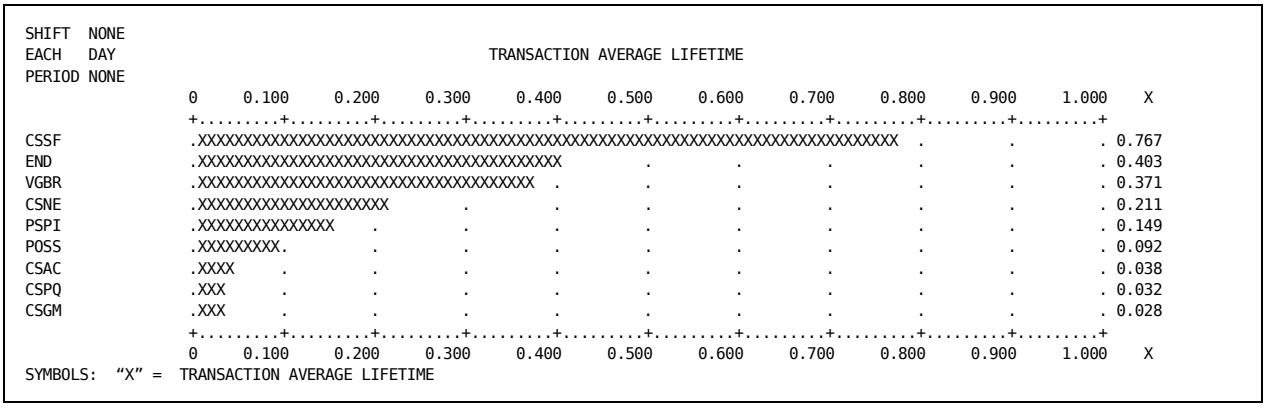
RANGE Command—Create Reports with a Specified Range

Use the RANGE command to restrict a report to resources whose activity falls within a specified range. The RANGE command is explained in further detail in the chapter “Commands.”

The following commands produce a graph of the lifetime of transactions with an average lifetime of 1.000 second or less. Because the DEFERRED operand is specified with the RANGE command, the range criterion check occurs after all data has been read. The Report Writer can then determine the average lifetime of each transaction.

```
GRAF TRAN LIFETIME SCALE(1.000)
  FROM 12/08/10 10:00:00
  TO   12/08/10 18:00:00
  RANGE TRAN LIFE 0.000-1.000 DEFERRED
END
RUN
```

A report like the following is generated:



```
SHIFT  NONE
EACH   DAY                                    TRANSACTION AVERAGE LIFETIME
PERIOD NONE
                  0     0.100     0.200     0.300     0.400     0.500     0.600     0.700     0.800     0.900     1.000    X
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
CSSF              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  .         .         . 0.767
END               .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX        .         .         .         .         .         .         . 0.403
VGBR              .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  .         .         .         .         .         .         . 0.371
CSNE              .XXXXXXXXXXXXXXXXXXX        .         .         .         .         .         .         .         . 0.211
PSPI              .XXXXXXXXXXXXXXX    .         .         .         .         .         .         .         .         . 0.149
POSS              .XXXXXXXXX.         .         .         .         .         .         .         .         .         . 0.092
CSAC              .XXXX     .         .         .         .         .         .         .         .         .         . 0.038
CSPQ              .XXX      .         .         .         .         .         .         .         .         .         . 0.032
CSGM              .XXX      .         .         .         .         .         .         .         .         .         . 0.028
                  +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                  0     0.100     0.200     0.300     0.400     0.500     0.600     0.700     0.800     0.900     1.000    X
SYMBOLS:  “X” =  TRANSACTION AVERAGE LIFETIME
```

GRAF2 Command Restrictions

Use the GRAF2 command with the GRAF command. Be sure that the variables you specify with the GRAF and GRAF2 commands are logically related.

Examples: Correct and Incorrect Usage of GRAF and GRAF2

- Transaction Use

  Correct:

  ```
  GRAF  TRANSACTION USE
  GRAF2 TERMINAL IO
  ```

  Incorrect:

  ```
  GRAF  DATASET REQUESTS
  GRAF2 TRANSACTION USE
  ```

- Terminal Response

  Correct:

  ```
  GRAF  TERMINAL RESP
  GRAF2 TRANSACTION USE
  ```

  Incorrect:

  ```
  GRAF  DATASET SERVTIME
  GRAF2 PROGRAM USE
  ```

- Transaction Use

  Correct:

  ```
  GRAF  TRANSACTION USE
  GRAF2 VARS TERM USE AND,
             PROGRAM USE AND,
             TRANSIENT USE
  ```

  Incorrect:

  ```
  GRAF  TERMINAL IO
  GRAF2 VARS TERM USE AND,
             PROGRAM USE AND,
             TRANSIENT USE
  ```

- Terminal IO

  Correct:

  ```
  GRAF  TERMINAL IO
  GRAF2 VARS TERM USE AND,
             TERM RESP AND,
             TERM MAXRESP AND,
             TERM BYTESREAD
  ```

  Incorrect:

  ```
  GRAF  TERMINAL IO
  GRAF2 VARS TERM USE AND,
             TRANSACTION USE
  ```

FOR Operand—Limit the Number of Identifiers in a GRAF2 Report

Use the FOR *n* operand to produce a report on the *n* resources with the highest or lowest values for the variable being graphed. The highest or lowest values depend on the sorting order. You can specify the FOR operand only with the first variable you specify with the GRAF2 command.

The following commands produce a report showing:

- The four transactions with the longest lifetimes
- The three terminals with the highest response times for each of those four transactions

The character *F* is used to graph transaction life and *R* is used to graph terminal response time.

```
GRAF  TRAN LIFE FOR 4 CHAR1(F)
 GRAF2 TERM RESP FOR 3 CHAR1(R)
 END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY                                        TRANSACTION AVERAGE LIFETIME
PERIOD NONE
                0     51.36     102.7     154.0     205.4     256.8     308.1     359.5     410.9     462.2     513.6    F
                0     0.225     0.449     0.674     0.899     1.123     1.348     1.573     1.798     2.022     2.247        R
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
PINV            .FFFF
```

SCALE Operand—Change the Scale of a GRAF2 Report

Use the SCALE operand to specify a scale for the GRAF2 variable. This operand is useful for creating a common scale in reports on time-related variables such as variables for transaction lifetime and terminal response time.

The following commands include the SCALE operand to produce a report that uses the same scale for both the GRAF and GRAF2 variables. The default plot characters are used:

```
GRAF  TRANSACTION LIFE FOR 3 SCALE(100.0)
 GRAF2 TERMINAL    RESP FOR 2 SCALE(100.0)
 END
RUN
```

A report like the following is generated:

```
                                                                                                                X
                0        10        20        30        40        50        60        70        80        90       100          *
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
PVCK            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX         . 91.40
  L401          .*************************************************************************************************>       168.7
  L205          .*************************************************************************************************>       110.
 JOUR            .XXXXXXXXXX         .         .         .         .         .         .         .         .         . 10.09
  L205          .************       .         .         .         .         .         .         .         .         .       11.65
  L702          .**********         .         .         .         .         .         .         .         .         .       10.13
PIAF            .XXXX     .         .         .         .         .         .         .         .         .         . 4.461
  L601          .**********         .         .         .         .         .         .         .         .         .       9.968
  L703          .**********         .         .         .         .         .         .         .         .         .       9.828
  L407          .******** .         .         .         .         .         .         .         .         .         .       8.237
                 +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                 0        10        20        30        40        50        60        70        80        90       100          *
                                                                                                                  X
SYMBOLS:  “X” =  TRANSACTION LIFETIME IN SECONDS.
          “*” =  TERMINAL RESPONSE TIME IN SECONDS.
```

GRAF2 TRANNUM—GRAF2 Command with an Identifier

When you specify a single variable with the GRAF2 command, you can specify an ID option with the GRAF2 command. Specifying an ID option will graph by an identifier other than the default.

The following commands produce a report on the three users with the highest terminal response times. For each user, terminal response times for the five transactions with the highest terminal response times are also graphed. If the ID option TRANNUM had not been specified with the GRAF2 command, the graphs would have been by terminal ID. Terminal ID is the default for terminal variables.

```
GRAF  USERID  TERM RESP FOR 3 SCALE(2.000) CHAR1(U)
 GRAF2 TRANNUM TERM RESP FOR 5 SCALE(2.000) CHAR1(T)
 END
RUN
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY                                    TERMINAL AVERAGE RESPONSE TIME
PERIOD NONE
                0     0.200     0.400     0.600     0.800     1.000     1.200     1.400     1.600     1.800     2.000    O
                0     0.889     1.777     2.666     3.554     4.443     5.332     6.220     7.109     7.997     8.886          T
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
HYZHE01         .UUUU
```

A report like the following is generated:

```
SHIFT  NONE
EACH   DAY
PERIOD NONE
                0     10.00     20.00     30.00     40.00     50.00     60.00     70.00     80.00     90.00     100.0    X
                0         5        10        15        20        25        30        35        40        45        50         *
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
SPAR            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 99.88
  FILE USE      .*****************************************************************   .         .         .         .         33
  TERM IO       .****************************************       .         .         .         .         .         .         21
  TRAN WAITS    .*************************************************************************************************     .         47
PDDR            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  .         .         .         .         .         .         . 37.29
  FILE USE      .********************************************************* .         .         .         .         .         29
  TERM IO       .*****************************************************   .         .         .         .         .         28
  TRAN WAITS    .************************************************************      .         .         .         .         3
DEDA            .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  .         .         .         .         .         .         . 36.58
  FILE USE      .*************************************************************************************************>         76
  TERM IO       .*************************************************************************************************>         64
  TRAN WAITS    .*************************************************************************************************>         90
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
                0         5        10        15        20        25        30        35        40        45        50         *
                0     10.00     20.00     30.00     40.00     50.00     60.00     70.00     80.00     90.00     100.0    X
SYMBOLS:  "X" = TRANSACTION AVERAGE LIFETIME
```

Commands to Graph Transactions

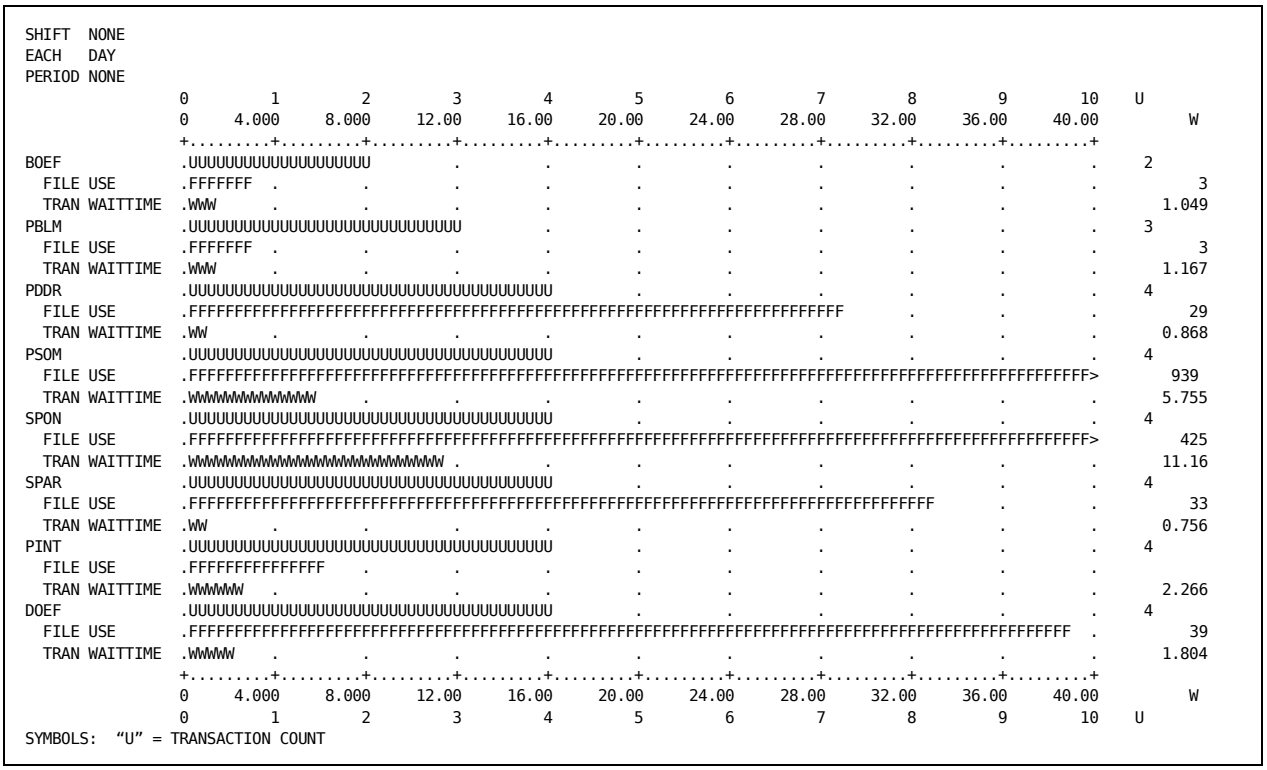
The following commands graph the transaction use of the eight transactions used the least, and, for each:

- The number of files used
- The amount of transaction waits per use

The CHAR1 operands specify a different plot character for each variable.

```
GRAF       TRAN USE  CHAR1(U) SCALE(10) FOR 8 SORT ASCENDING
GRAF2 VARS FILE USE  CHAR1(F) SCALE(40) AND,
           TRAN WAIT CHAR1(W)
END
RUN
```

A report like the following is generated:



```
SHIFT  NONE
EACH   DAY
PERIOD NONE
                0         1         2         3         4         5         6         7         8         9        10    U
                0     4.000     8.000     12.00     16.00     20.00     24.00     28.00     32.00     36.00     40.00          W
                +.........+.........+.........+.........+.........+.........+.........+.........+.........+.........+
BOEF            .UUUUUUUUUUUUUUUUUUU         .         .         .         .         .         .         .         .     2
  FILE USE      .FFFFFFF  .         .         .         .         .         .         .         .         .         .         3
  TRAN WAITTIME .WWW      .         .         .         .         .         .         .         .         .         .     1.049
PBLM            .UUUUUUUUUUUUUUUUUUUUUUUUUUUUU         .         .         .         .         .         .         .     3
  FILE USE      .FFFFFFF  .         .         .         .         .         .         .         .         .         .         3
  TRAN WAITTIME .WWW      .         .         .         .         .         .         .         .         .         .     1.167
PDDR            .UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU         .         .         .         .         .         .     4
  FILE USE      .FFFFFF
```

Chapter 11: Historical Reports

This section contains the following topics:

About Historical Reports (see page 261)
Output Considerations (see page 262)
FLASHBACK LIST Command (see page 262)

About Historical Reports

You can create historical reports using the FLASHBACK LIST command. Historical, or flashback, reports are tabular reports that contain detailed information about transactions. The FLASHBACK LIST generates one line of output per transaction.

Transactions are presented in chronological order of ending time. You cannot sort flashback reports.

To create historical reports, review the following:

- Report Writer commands
- Descriptions of Report Writer variables and lists of the ID options you can use with each class of variable
- Descriptions of ID options and their use with commands and variables

Important! Each flashback report must be run as a single report in a separate run. In other words, you cannot specify another primary report command (such as TAB, PLOT, or GRAF) in a job stream with a FLASHBACK LIST command.

More information:

Commands (see page 39)
Variables (see page 109)
ID Options (see page 167)

Output Considerations

The FLASHBACK commands can generate enormous amounts of output when run against a large input file. To avoid problems, use either the FROM and TO commands, the SELECT command, or the COUNT command to limit output.

When you do not use the FROM and TO commands, the corresponding FROM and TO fields in the printout page headers are not utilized. Also, the values are set to the default dates and times (01/01/00 00.00.00, 12/31/99 23.59.59).

The way the FLASHBACK reports process records prevents from enumerating the date and time range of the records.

FLASHBACK LIST Command

Use the FLASHBACK LIST command to produce a tabular report containing one line of output per transaction or event.

To produce default FLASHBACK reports for various systems, use the following syntax:

```
FLASHBACK [LIST] system
```

system

Designates one of the following systems:

- CICS
- IMS
- AUDITLOG
- MVS

Default CICS FLASHBACK LIST Report

To create a default CICS flashback report, specify the following command without any ID options or variables:

```
FLASHBACK LIST
```

Use either the COUNT command or the FROM and TO commands to limit report output.

The following commands produce the default FLASHBACK LIST or FLASHBACK CICS report. The FROM and TO commands limit the report to a short time period.

```
FLASHBACK LIST
FROM 12/06/08 21:22:00
  TO 12/06/08 21:24:00
END
RUN
```

A report like the following is generated:

CICS ID	END TIME	TERM ID	OPER ID	TRAN ID	TRAN NUMBER	TRAN LIFE	TRAN CPU	TERM RESP	TERM IO	TRAN I/O	FILE COUNT	FILE USE	PROG COUNT	ABEND CODE
B1	21.22.41	N/A	N/A	TCP	0	911.9	2.110	0.000	0	0.381	1	241	11	
B1	21.22.42	N/A	N/A	KCP	0	912.3	1.878	0.000	0	0.000	0	0	0	
B1	21.23.07	N/A	N/A	JJJ	0	937.1	1.942	0.000	0	0.000	0	0	0	
B1	21.23.07	N/A	N/A	CATD	51	6.529	0.326	0.000	0	0.000	0	0	5	
B1	21.23.08	N/A	N/A	CSNE	52	0.081	0.068	0.000	0	0.000	0	0	3	
B1	21.23.09	ZA00	N/A	CSGM	53	0.264	0.010	0.013	2	0.000	0	0	1	
B1	21.23.15	N/A	N/A	VGBR	55	0.332	0.023	0.000	0	0.000	0	0	2	
B1	21.23.21	N/A	N/A	CSNE	56	0.013	0.002	0.000	0	0.000	0	0	3	
B1	21.23.41	X250	N/A	VGBQ	57	19.33	0.151	0.389	5	0.000	0	0	7	AKCT
	21.23.41				324	309.7	0.723	0.201	7	0.042	0	241	4	

The sample report contains the following information:

CICS ID

Specifies the region ID in which the data was collected.

END TIME

Specifies the time the transaction completed.

TERM ID

Specifies the ID of the terminal from which the transaction was executed,

TRAN ID

Specifies the transaction ID.

TRAN NUMBER

Specifies the transaction number.

TRAN LIFE

Specifies the transaction lifetime (seconds).

TRAN CPU

Specifies the transaction CPU time (seconds).

TERM RESP

Specifies the terminal average response time (seconds).

TERM IO

Specifies the number of terminal I/Os by the transaction.

TRAN I/O

Specifies the total I/O time for the transaction.

FILE COUNT

Specifies the number of files accessed by the transaction.

FILE USE

Specifies the number of file I/Os by the transaction.

PROG COUNT

Specifies the number of programs used by the transaction.

ABEND CODE

Specifies the abend code for the transaction, if any.

Create a Default FLASHBACK IMS Report

The FLASHBACK IMS reports provide you with detailed information about IMS transactions.

To create a default IMS flashback report, specify the following command without any ID options or variables:

```
FLASHBACK IMS
```

The same rules apply as for the other default reports, for example, the default CICS FLASHBACK LIST Report.

The sample report contains following information:

DATE

Specifies the date when the audit event occurred.

TIME

Specifies the time when the audit event occurred.

TRAN ID

Specifies the transaction ID.

PSB NAME

Specifies the program specification block.

PST NUM

Specifies the partition specification table number.

LTERM

Specifies the logical terminal name.

REGION NAME

Specifies the region name.

PROC TYPE

Specifies the transaction processing type.

TRAN LIFE

Specifies the transaction lifetime in seconds.

TRAN INQUEU

Specifies the input queue time in seconds.

TRAN OUTQUEU

Specifies the output queue time in seconds.

TRAN PROC

Specifies the transaction processing time in seconds.

More information:

Default CICS FLASHBACK LIST Report (see page 262)

Create a Default FLASHBACK AUDITLOG Report

To create a default audit flashback report, specify the following command without any ID options or variables:

```
FLASHBACK AUDITLOG
```

Use either the COUNT or FROM and TO commands to limit report output.

The following commands produce the default FLASHBACK AUDITLOG report. The FROM and TO commands limit the report to a short time period.

```
FLASHBACK AUDITLOG
FROM 05/31/09 20.04.01
TO   05/31/09 20.06.00
END
RUN
```

A report like the following is generated:

```
                    AUDIT        SYSTEM   SUBSYS    JOB      TASK   SECURITY PRODUCT    USER   TERMINAL
  DATE     TIME      TYPE          NAME     ID      ID       NAME    GROUP   RELEASE     ID       ID
-------- -------- ---------------- -------- -------- -------- -------- -------- -------- -------- --------
05/31/09 20.04.29 Session Logon    CA31       GSVX   TSU53679          DEFAULT    12.0   MOLCE01  A55T
05/31/09 20.04.46 Session Logoff   CA31       GSVX   TSU53679          DEFAULT    12.0   MOLCE01  A55T
05/31/09 20.04.55 Session Logon    CA31       GSVX   TSU53463          DEFAULT    12.0   ROGRO02  A55T
05/31/09 20.05.00 Session Logon    CA31       GSVX   STC47269          ADMIN      12.0   STCSYS
05/31/09 20.05.00 Session Logon    CA31       GSVX   STC47268          ADMIN      12.0   STCSYS
05/31/09 20.05.00 Session Logoff   CA31       GSVX   STC47269          ADMIN      12.0   STCSYS
05/31/09 20.05.02 Session Logoff   CA31       GSVX   STC47268          ADMIN      12.0   STCSYS
05/31/09 20.05.16 JES2 Job Modify  CA31       GSVX   TSU53463          DEFAULT    12.0   ROGRO02  A55T
05/31/09 20.05.16 Console command  CA31       GSVX   TSU53463          DEFAULT    12.0   ROGRO02  A55T
05/31/09 20.05.39 Session Logon    CA31       GSVX   TSU53679          DEFAULT    12.0   MOLCE01  A55T
-------- -------- ---------------- -------- -------- -------- -------- -------- -------- -------- --------
05/31/09 20.05.39
-------- -------- ---------------- -------- -------- -------- -------- -------- --------------  - --------
```

The sample report contains following information:

DATE

Specifies the date when audit event occurred.

TIME

Specifies the time when audit event occurred.

AUDIT TYPE

Specifies the audit record type.

SYSTEM NAME

Specifies the name of the system where audit event occurred.

SUBSYS ID

Specifies the name of subsystem.

JOB ID

Specifies the ID of the job that caused the event.

TASK NAME

Specifies the task name.

SECURITY GROUP

Specifies the name of a security group to which a user or program that caused an audit event belongs.

PRODUCT RELEASE

Specifies the product release.

USER ID

Specifies the ID of user that causes an audit event.

TERMINAL ID

Specifies the ID of terminal used by user that causes an event.

Limit FLASHBACK Reports to Specified Resources

You can limit the information contained in a flashback report to specific resources, such as a single operator or transactions beginning with the letters *ACCT*. To limit the information, *s*pecify an ID option and identifiers with the FLASHBACK LIST or FLASHBACK *system* command.

Example: Using a Single ID Option and Identifier

The following commands produce a FLASHBACK LIST report showing only the transactions of the terminal SWS:

```
FLASHBACK LIST TERMID(0A02)
END
RUN
```

A report like the following is generated:

CICS ID	END TIME	TERM ID	TRAN ID	TRAN NUMBER	TRAN LIFE	TRAN CPU	TERM RESP	TERM IO	TRAN I/O	FILE COUNT	FILE USE	PROG COUNT	ABEND CODE
B1	00.45.29	0A02	POSS	67	1.011	0.274	1.011	2	0.000	0	0	3	
B1	00.45.35	0A02	POSS	69	2.387	0.051	1.187	3	0.199	1	2	6	
B1	00.45.38	0A02	POSS	70	0.419	0.249	0.000	1	0.000	0	0	3	
B1	00.45.44	0A02	DEDA	71	5.040	0.705	0.992	4	0.034	1	1	4	
B1	00.45.44	0A02	POSS	72	0.078	0.023	0.034	2	0.039	1	1	4	
B1	00.46.03	0A02	POSS	73	0.025	0.002	0.025	2	0.000	0	0	3	
B1	00.46.12	0A02	SPAM	75	1.422	0.291	0.462	3	0.040	1	1	4	
B1	00.46.18	0A02	SPAM	76	0.446	0.085	0.199	3	0.052	1	1	4	
B1	00.46.37	0A02	SPAM	77	0.094	0.020	0.000	1	0.048	1	1	4	
B1	00.46.38	0A02	POSS	78	0.088	0.025	0.033	2	0.038	1	1	4	
B1	00.46.40	0A02	POSS	79	0.034	0.021	0.000	1	0.000	0	0	3	
B1	00.58.19	0A02	PBLM	80	699.2	1.925	0.109	156	0.030	1	1	4	
B1	00.58.19	0A02	POSS	81	0.091	0.026	0.041	2	0.047	1	1	4	
B1	05.28.09	0A02	POSS	574	0.028	0.017	0.000	1	0.000	0	0	3	
B1	05.28.09	0A02	PSPI	575	0.047	0.016	0.009	2	0.000	0	0	3	
B1	05.28.19	0A02	PSPI	577	0.177	0.054	0.044	4	0.000	0	0	3	
B1	05.28.29	0A02	PSPI	580	0.170	0.049	0.029	5	0.000	0	0	3	
B1	05.28.41	0A02	PSPI	581	0.152	0.050	0.024	5	0.000	0	0	3	
B1	05.28.53	0A02	PSPI	582	0.148	0.049	0.024	5	0.000	0	0	3	
B1	05.29.03	0A02	PSPI	584	0.154	0.049	0.025	5	0.000	0	0	3	
B1	05.29.20	0A02	PSPI	588	0.118	0.040	0.018	5	0.000	0	0	3	
B1	05.29.37	0A02	PSPI	592	0.100	0.045	0.013	5	0.000	0	0	3	
B1	05.29.48	0A02	PSPI	593	0.130	0.046	0.018	5	0.000	0	0	3	
B1	05.32.32	0A02	PSPI	603	0.219	0.051	0.039	5	0.000	0	0	3	
B1	05.34.14	0A02	PSPI	606	0.129	0.032	0.129	2	0.000	0	0	3	
	05.34.14			8003	28.47	0.168	0.132	231	0.021	0	10	3	

Example: Using Multiple ID Options

The following commands produce a transaction log:

```
FLASHBACK LIST REGID JOBNAME DATETIME TRANID,
                                      TERMID,
                                      TRAN NUM
END
RUN
```

A report like the following is generated:

```
                                        TRAN      TERM      TRAN
ID JOBNAME    DATE     TIME      ID       ID        NUMBER
-- -------- -------- --------- -------- -------- --------
A1 CICA     12/06/08  21.16.58  CSDA     CNSL          19
A1 CICA     12/06/08  21.21.52  CIRB     CNSL          20
B1 CICB     12/06/08  21.22.41  TCP      N/A            0
A1 CICA     12/06/08  21.24.20  TCP      N/A            0
B1 CICB     12/06/08  21.22.42  KCP      N/A            0
B1 CICB     12/06/08  21.23.06  JJJ      N/A            0
B1 CICB     12/06/08  21.23.06  CATD     N/A           51
B1 CICB     12/06/08  21.23.07  CSNE     N/A           52
B1 CICB     12/06/08  21.23.08  CSGM     ZA00          53
B1 CICB     12/06/08  21.23.15  VGBR     N/A           55
B1 CICB     12/06/08  21.23.20  CSNE     N/A           56
-- -------- -------- --------- -------- -------- --------
B1 CICB     12/06/08  21.23.20                       306
-- -------- -------- --------- -------- -------- --------
```

Create Report with Specified Columns

To produce a flashback report with different columns than the default report, specify one or more variables with the FLASHBACK LIST command. A column is created in the report for each variable you specify.

The following commands produce a report that displays all the components comprising transaction lifetime. TRAN EXPC and TRAN CPU are not true components of transaction lifetime, because the times found in these variables are accounted for in other variables.

```
FLASHBACK LIST TIME TRANID TRAN NUM     AND,
                           TRAN LIFE    AND,
                           TRAN USER    AND,
                           TRAN CICS    AND,
                           TRAN I/OTIME AND,
                           TRAN WAIT    AND,
                           TRAN SUSP    AND,
                           TRAN WTR     AND,
                           TRAN RSCS    AND,
                           TRAN EXPC    AND,
                           TERM I/OTIME AND,
                           TRAN CPU
  COUNT 20
END
RUN
```

A report like the following is generated:

```
          TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN     TRAN
  TIME      ID   NUMBER     LIFE     USER     CICS      I/O     WAIT     SUSP      WTR     RSCS     EXPC     TMIO      CPU
-------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
21.16.58 CSDA         19    0.016    0.000    0.000    0.000    0.000    0.000    0.001    0.000    0.001    0.000    0.011
21.21.52 CIRB         20    13.30    0.000    0.000    0.000    1.375    0.000    0.002    5.519    0.004    1.521    1.175
21.22.41 TCP           0    911.9    0.000    0.000    0.381    901.3    0.000    2.173    3.597    0.325    0.000    2.110
21.24.20 TCP           0    912.3    0.000    0.000    0.000    906.8    0.000    1.372    1.900    0.233    0.000    1.051
21.22.42 KCP           0    912.3    0.000    0.000    0.000    0.000    0.000    895.7    0.000    0.000    0.000    1.878
21.23.06 JJJ           0    937.1    0.000    0.000    0.000    0.418    934.7    0.001    0.000    0.001    0.000    1.942
21.23.06 CATD         51    6.529    0.000    0.000    0.000    0.124    0.000    0.009    2.579    0.003    0.000    0.326
21.23.07 CSNE         52    0.081    0.000    0.000    0.000    0.000    0.000    0.009    0.000    0.000    0.000    0.068
21.23.08 CSGM         53    0.264    0.000    0.000    0.000    0.000    0.000    0.002    0.139    0.001    0.017    0.010
21.23.15 VGBR         55    0.332    0.000    0.000    0.000    0.000    0.000    0.080    0.221    0.002    0.000    0.023
21.23.15 CSNE         56    0.013    0.000    0.000    0.000    0.000    0.000    0.002    0.000    0.000    0.000    0.002
-------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
21.23.15             306    335.8    0.000    0.000    0.035    164.5    84.97    81.76    1.269    0.052    0.140    0.782
-------- -------- -------- -------- -------- -------- -------- -------- -------- -------------- -- -------- -------- --------
```

Create a Customized FLASHBACK LIST Report

You can produce a FLASHBACK LIST report that has:

- The date, time, or day in the first column
- The date and time in the first two columns

To produce either of these reports, specify the DATE, TIME, DAY, or DATETIME operand with the FLASHBACK LIST command.

The width of the output page limits the number of columns that can appear in a report. Use the DATE, TIME, DAY, or DATETIME operand only if you specify the variables to include in your report and limit their number. The number of variables you can specify depends upon the width of the column required for each.

Example: Sample Commands

The following FLASHBACK LIST command is specified with the DATETIME operand. The COUNT command, specified with the SELECTED operand, limits the report to 15 transactions meeting either the FROM or the TO criterion specified.

```
FLASHBACK LIST DATETIME TRANID TRAN LIFE AND,
                        TRAN I/OTIME    AND,
                        TRAN WAIT       AND,
                        TRAN SUSP       AND,
                        TERM I/OTIME    AND,
                        TRAN CPU
  FROM 12/06/08 21:00:00
  TO   12/06/08 22:00:00
  COUNT 15 SELECTED
END
RUN
```

A report like the following is generated:

```
                  TRAN      TRAN      TRAN      TRAN      TRAN      TRAN      TRAN
  DATE     TIME      ID      LIFE       I/O      WAIT      SUSP      TMIO       CPU
-------- -------- -------- -------- -------- -------- -------- -------- --------
12/06/08 21.16.58 CSDA       0.016    0.000    0.000    0.000    0.000    0.011
12/06/08 21.21.52 CIRB      13.30     0.000    1.375    0.000    1.521    1.175
12/06/08 21.22.41 TCP       911.9     0.381    901.3    0.000    0.000    2.110
12/06/08 21.24.20 TCP       912.3     0.000    906.8    0.000    0.000    1.051
12/06/08 21.22.42 KCP       912.3     0.000    0.000    0.000    0.000    1.878
12/06/08 21.23.06 JJJ       937.1     0.000    0.418    934.7    0.000    1.942
12/06/08 21.23.06 CATD      6.529     0.000    0.124    0.000    0.000    0.326
12/06/08 21.23.07 CSNE      0.081     0.000    0.000    0.000    0.000    0.068
12/06/08 21.23.08 CSGM      0.264     0.000    0.000    0.000    0.017    0.010
12/06/08 21.23.15 VGBR      0.332     0.000    0.000    0.000    0.000    0.023
12/06/08 21.23.41 VGBQ      19.33     0.000    0.000    0.000    18.40    0.151
12/06/08 21.28.18 VGBS      306.6     0.000    0.000    0.000    304.6    0.633
12/06/08 21.28.20 CSAC      0.647     0.000    0.000    0.000    0.000    0.015
12/06/08 21.28.23 END       0.486     0.000    0.000    0.000    0.000    0.232
12/06/08 21.28.23 CSNE      0.020     0.000    0.000    0.000    0.000    0.003
-------- -------- -------- -------- -------- -------- -------- -------- --------
12/06/08 21.28.23           251.3     0.024    113.1    58.42    20.29    0.602
-------- -------- -------- -------- -------- -------- -------- -------- --------
```

Limit Default FLASHBACK Reports to Specified Variable Values

The RANGE command with FLASHBACK LIST or FLASHBACK *system* command lets you produce a report containing resources in specified range criteria. You can specify more than one RANGE command in a single report.

The following commands produce a report showing all transactions that had an average terminal response time greater than 10 seconds:

```
FLASHBACK LIST
  RANGE TERM RESP 10+ DEFERRED
END
RUN
```

A report like the following is generated:

```
CICS   END      TERM     OPER     TRAN     TRAN     TRAN     TRAN     TERM     TERM     TRAN     FILE     FILE     PROG    ABEND
 ID    TIME      ID       ID       ID     NUMBER     LIFE      CPU     RESP       IO      I/O     COUNT     USE     COUNT    CODE
---- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
 B1  23.25.13 N/A      N/A                  276    56.41    0.007    28.19        2    0.000        0        0        1
 A1  23.25.02 N/A      N/A                   27    44.26    0.065    21.65        2    0.000        0        0        1
 A1  01.47.08 WAK      N/A                  278    10.92    0.317    10.92        2    0.000        0        0        7
 A1  01.48.24 WAK      N/A                  279    15.99    0.060    15.99        2    0.000        0        0        4
 B1  02.11.44 N/A      N/A                 3498    14.20    0.178    14.20        2    0.106        1        1        8
 B1  02.14.43 AFL      N/A                 3708    13.18    0.062    13.18        2    0.000        0        0        4
 A1  02.23.35 WAS      N/A                  794    24.26    0.059    24.26        2    0.000        0        0        4
 A1  02.41.37 AFK      N/A                 1157    22.84    0.654    16.20        3    0.339        2      114        5
 B1  02.41.48 N/A      N/A                 5247    12.28    1.820    12.28        2    5.885        2      432        8
 A1  02.42.11 PSH      N/A                 1164    36.29    5.162    14.60        4    0.000        0        0        7
 B1  02.44.43 N/A      N/A                 5411    14.43    0.139    14.43        2    0.046        1        1        8
 B1  02.49.25 N/A      N/A                 5671    38.82    5.740    38.82        2    23.64        3      618        1
---- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
---- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- ----------- -------- -------- --------
```

Chapter 12: Messages

This section contains the following topics:

Syntax Error Messages (see page 275)
Other Messages (see page 275)
ERPT200I (see page 276)
ERPT201E (see page 276)
ERPT202E (see page 277)
ERPT204E (see page 277)
ERPT205E (see page 277)
ERPT206E (see page 278)
ERPT207E (see page 278)
ERPT208E (see page 278)
ERPT210E (see page 279)
ERPT211E (see page 279)
ERPT212E (see page 279)
ERPT213E (see page 280)
ERPT214E (see page 280)
ERPT215E (see page 281)
ERPT216E (see page 281)
ERPT217E (see page 281)
ERPT218E (see page 282)
ERPT219E (see page 282)
ERPT221W (see page 283)
ERPT222W (see page 283)
ERPT223E (see page 283)
ERPT224I (see page 284)
ERPT226E (see page 284)
ERPT227E (see page 285)
ERPT228I (see page 285)
ERPT229W (see page 285)
ERPT230I (see page 286)
ERPT231I (see page 286)
ERPT232I (see page 286)
ERPT233I (see page 286)
ERPT234E (see page 287)
ERPT235E (see page 287)
ERPT236E (see page 287)
ERPT237I (see page 288)
ERPT238E (see page 288)
ERPT239E (see page 288)
ERPT240E (see page 289)
ERPT242I (see page 289)
ERPT244E (see page 289)
ERPT247E (see page 290)
ERPT254E (see page 290)

ERPT255E (see page 290)
ERPT257I (see page 291)
ERPT258I (see page 291)
ERPT259I (see page 291)
ERPT260I (see page 292)
ERPT261E (see page 292)
ERPT262E (see page 292)
ERPT263E (see page 293)
ERPT264E (see page 293)
ERPT265E (see page 293)
ERPT266E (see page 294)
ERPT268I (see page 294)
ERPT269I (see page 294)
ERPT271E (see page 295)
ERPT272W (see page 295)
ERPT273I (see page 295)
ERPT274I (see page 296)
ERPT275E (see page 296)
ERPT276E (see page 296)
ERPT277E (see page 297)
ERPT278I (see page 297)
ERPT280E (see page 297)
ERPT281W (see page 298)
ERPT282I (see page 298)
ERPT286E (see page 298)
ERPT287E (see page 299)
ERPT289I (see page 299)
ERPT290E (see page 299)

Syntax Error Messages

Messages issued by the Report Writer begin with the letters ERPT. Some Report Writer messages identify syntax errors in your history reporting job stream. These messages appear in the output from the Report Writer. Such messages identify syntax errors in the following way:

- The line in the job stream that contains the syntax error is displayed.
- The syntax error is marked on the following line with a dollar sign ($).
- The message identifying the syntax error appears on the line below the line containing the dollar sign.

Syntax errors identified by messages of type E cause the Report Writer to terminate.

The following example shows how the Report Writer identifies a syntax error:

```
TAB DATE MVS CPU %BUSY AND,
         MVS JOB CONT
                 $
ERPT204E ILLEGAL VARIABLE
END
RUN
ERPT202E 1 SYNTAX ERRORS ENCOUNTERED
ERPT229I RUN TERMINATED FOR ABOVE REASONS
```

In this example, MVS JOB CONT is identified as an illegal variable. The Report Writer terminates.

Other Messages

The following are the messages issued by the Report Writer.

ERPT200I

XPFRMAIN *v.rm status mm/dd/yy hh.mm.ss*

Reason:

The Report Writer has begun or has completed running the reports you requested.

status

Indicates either INITIATED or COMPLETED.

v.rm

Provides the version, release, and maintenance level of the Report Writer.

mm/dd/yy hh.mm.ss

Provides the date and time at which the Report Writer has begun or has completed running the reports.

Action:

None.

ERPT201E

MAIN STORAGE EXHAUSTED, PGM -program, OFFSET -offset

Reason:

The program identified by *program* requested storage for the Report Writer to run, but not enough virtual storage is available.

offset

Provides the offset within the program that requested storage.

Action:

Do one or more of the following:

- Reduce the number of plot lines that the Report Writer prints.

  Note: The need for virtual memory by the Report Writer is nearly proportional to the number of plot lines that are to print.
- Specify fewer reports at a time.
- Specify a larger interval in the EACH command.
- Specify a shorter time range using the FROM and TO commands.
- Use the COUNT command.
- Run the job in a larger partition or region.

ERPT202E

_n_ SYNTAX ERRORS ENCOUNTERED

Reason:

The Report Writer has encountered *n* syntax errors.

n

Specifies the number of syntax errors.

Action:

Correct the syntax errors and resubmit the job.

ERPT204E

ILLEGAL VARIABLE

Reason:

This message identifies a syntax error. The variable above the dollar sign ($) is invalid.

Action:

Correct the invalid variable.

More information:

Syntax Error Messages (see page 275)

ERPT205E

UNKNOWN COMMAND

Reason:

This message identifies a syntax error. The control statement above the dollar sign ($) contains an invalid command.

Action:

Correct the invalid command.

More information:

Syntax Error Messages (see page 275)

ERPT206E

DUPLICATE COMMANDS

Reason:

This message identifies a syntax error. The command above the dollar sign ($) is a duplicate. You can use some commands only once in any given report. For instance, only one FROM or TO command can be used in a single report.

Action:

Remove the duplicate command.

More information:

Syntax Error Messages (see page 275)

ERPT207E

TOO FEW OPERANDS

Reason:

The Report Writer did not find enough operands to produce a report. The continuation character could be missing at the end of a line. The continuation comma tells the Report Writer that more operands follow on the next line. A dollar sign ($) underscores the last operand that the Report Writer found.

Action:

Either add more operands or place a comma after the last character on the line.

More information:

Syntax Error Messages (see page 275)

ERPT208E

INVALID OPERAND

Reason:

This message identifies a syntax error. The operand above the dollar sign ($) is incomplete or invalid.

Action:

Correct the partial or invalid operand.

More information:

Syntax Error Messages (see page 275)

ERPT210E

UNEXPECTED OPERANDS: IGNORED

Reason:

This message identifies a syntax error. You exceeded the maximum number of operands the current command can accept. The operand above the dollar sign ($) is the first operand to exceed the maximum number.

Action:

Delete the excessive operands and rerun the report.

More information:

Syntax Error Messages (see page 275)

ERPT211E

OPERAND IGNORED

Explanation:

This message follows other syntax error messages and indicates that the current operand has been discarded. The Report Writer continues to look at the remainder of the operands to finish the error check and then terminates processing.

Action:

Correct the syntax error.

ERPT212E

MISSING COMMA OR PARENTHESIS

Explanation:

This message identifies a syntax error. You omitted a comma or parenthesis at the location above the dollar sign ($).

Action:

Insert a comma or a parenthesis.

More information:

Syntax Error Messages (see page 275)

ERPT213E

EACH VALUE GREATER THAN PERIOD VALUE

Reason:

The EACH time value is greater than the PERIOD time value.

The EACH command defines the timespan represented by each detail line The PERIOD command defines the timespan covered by a Report Writer summary. Therefore, the EACH time value cannot be greater than the PERIOD time value.

Action:

Change the EACH or PERIOD value to correct the error. The default for the EACH command is one day.

ERPT214E

RUN CARD WITHOUT PREVIOUS END CARD

Reason:

This message identifies a syntax error. You tried to issue a RUN command before issuing an END command to end the last report. A dollar sign ($) underscores the RUN command that you issued out of sequence.

Action:

Enter an END command to end the last report before using the RUN command.

More information:

Syntax Error Messages (see page 275)

ERPT215E

COMMAND IGNORED

Reason:

This message identifies a syntax error. The Report Writer ignored the current command and all of its operands. This error message follows other error messages that identify the specific problem. A dollar sign ($) underscores the part of the command that generated the error.

Action:

Correct the command and resubmit the report.

More information:

Syntax Error Messages (see page 275)

ERPT216E

END CARD ENTERED WITHOUT ANY REPORT COMMANDS

Reason:

An END command signaled the end of a report, but no report existed.

Action:

Specify a report before you issue the END command.

ERPT217E

NO MAIN REPORT COMMAND FOR THIS REPORT

Reason:

The END command has signaled the end of a report, but the Report Writer did not find a primary command to begin the report.

Action:

Specify a primary command for the report.

ERPT218E

INTERNAL ERROR: ADDRESS - *addr* CSECT - *cname* OFFSET - *offset*

Reason:

An internal error occurred.

addr

Identifies the address where the error occurred.

cname

Specifies the name of the CSECT.

offset

Specifies is the offset into the CSECT.

Action:

Collect as much information about the problem as possible, including the listing of the run in which this error occurred and then contact Technical Support.

ERPT219E

EOF WITH MISSING END COMMAND

Reason:

End-of-file was reached, but the required END command was missing for the last report.

Action:

End this report and all other reports with the END command.

ERPT221W

NO MATCHING DATA

Reason:

This message occurs during report writing rather than during syntax checking. The Report Writer could find no data in the log file to match the following items you specified: the times, the identifiers, or the variables with non-zero activity.

The Report Writer considers zero activity for a given variable to be no activity.

For example, you requested a report on the variable TRANSACTION WAITS, and no waits existed for any transactions during the time you specified. You would then receive this message.

Action:

Either modify your commands to change the time boundaries for the report or make the identifier restrictions more general.

ERPT222W

NO REPORTS TO PRINT

Reason:

You tried to execute the Report Writer but did not specify any report commands.

Action:

Specify one or more report commands.

ERPT223E

XPFRMAIN ABORTED FOR THE ABOVE REASONS

Reason:

Because of the errors listed above this message, your reports were not printed.

Action:

Review the messages preceding this message and then take the appropriate action. If you need help, contact Technical Support.

ERPT224I

***format* DATE FORMAT REQUESTED**

Reason:

The Report Writer expects the date format identified by format, but your report includes dates in another format.

format

Specified as either dd/mm/yy or mm/dd/yy (where dd = day, mm = month, and yy = year).

Action:

Alter your dates to conform to the format the Report Writer expects. You can use the DATEFRMT operand of the OPTION command to specify the type of date format you want.

More information:

Commands (see page 39)

ERPT226E

NO LIST AFTER OPERAND

Reason:

This message identifies a syntax error. An ID option either has no identifiers or has no left parenthesis before its identifiers. The format for ID options requiring identifiers is a left parenthesis followed by one or more identifiers separated by commas, and a closing right parenthesis. One or more identifiers or a left parenthesis appear at the location above the dollar sign ($).

Action:

Supply the missing identifiers or parentheses.

More information:

Syntax Error Messages (see page 275)

ERPT227E

INCOMPATIBLE REPORT TYPES MIXED

Reason:

One of the following has occurred:

- The three report types PLOT, GRAF, and TAB are mutually exclusive. You have requested more than one type of these reports between END commands.
- You have specified the command EACH RECORD with the HPLOT command but have not included a SCALE operand with the HPLOT command.

Action:

Submit only one report type for each END command, or include a SCALE operand when you use the EACH RECORD command with the HPLOT command.

ERPT228I

QUIT COMMAND ENCOUNTERED

Reason:

A QUIT command was encountered. The QUIT command stops syntax checking and halts the Report Writer without reading another command.

Action:

No action is required.

ERPT229W

RUN TERMINATED FOR ABOVE REASONS

Reason:

The Report Writer terminated for the reasons listed above this message.

Action:

Correct the errors and resubmit the report.

ERPT230I

RUN COMPLETED SUCCESSFULLY

Reason:

The Report Writer successfully completed the reports you requested.

Action:

None.

ERPT231I

NO ERRORS FOUND IN CONTROL CARDS

Reason:

The Report Writer finished parsing your commands and found no errors.

Action:

None.

ERPT232I

filename OPENED

Reason:

The Report Writer successfully opened the file identified by filename.

Action:

None.

ERPT233I

filename CLOSED

Reason:

The Report Writer successfully closed the file identified by filename.

Action:

None.

ERPT234E

UNBALANCED PARENTHESIS

Reason:

This message identifies a syntax error. The number of left parentheses in your job stream does not match the number of right parentheses. A parenthesis appears at the location above the dollar sign ($).

Action:

Balance the number of left and right parentheses.

More information:

Syntax Error Messages (see page 275)

ERPT235E

NOT ENOUGH ROOM ON PAGE

Reason:

You are constructing a table with the TAB command, but the Report Writer has run out of space on the page for the columns you are requesting.

Action:

Remove some of the variables, decrease the size of the headings you selected with the HDR1 and HDR2 commands, or use the FORMAT variable option to reduce the number of columns used by each variable.

ERPT236E

'AND' NOT FOLLOWED BY A VARIABLE

Reason:

The Report Writer did not find a variable after the AND operand. The cause of this error could be the lack of a continuation comma after the AND operand.

Action:

Either add more variables or place a comma after the AND operand.

ERPT237I

SAMPLE REPORT EXPANSION

Reason:

The Report Writer generates the statements that follow this message.

Action:

None.

ERPT238E

INVALID VARIABLE/ID-OPTION COMBINATION

Reason:

The cause can be either of the following:

- The order of the variables in a multiple-variable report is incorrect
- You specified variables that are not logically related in a tabular report broken down by identifier

Action:

Change the order of your variables or specify only variables that are logically related.

ERPT239E

VSAM ERROR WAS DETECTED IN FILE filename REQUEST=xx.

Reason:

An attempt to access the file identified by filename failed.

xx

The VSAM request type.

Message ERPT240E follows this message.

Action:

Use the information in message ERPT240E to determine the problem.

ERPT240E

R15=X 'code1' ERROR CODE=X 'code2'

Reason:

Your attempt to access the file identified by filename in message ERPT239E failed. code1 is the return code from register 15, and code2 is the error code.

Action:

Use the codes in the message to correct the error. For the meanings of these codes, see the IBM documentation containing VSAM messages and codes.

ERPT242I

RECORD IGNORED, OUTSIDE TIME LIMITS

Reason:

The record that the Report Writer is currently processing is outside the limits set by the report parameters. The record is ignored.

Action:

None.

ERPT244E

INVALID LOG FILE RECORD FORMAT

Reason:

The record read by the Report Writer has an invalid record format.

Action:

Verify that the records you have instructed the Report Writer to read have been correctly identified to the Report Writer. Verify that the associated INPUT commands are correct.

More information:

Commands (see page 39)

ERPT247E

INVALID IDENTIFIER SPECIFIED

Reason:

This message identifies a syntax error. The identifier above the dollar sign ($) is invalid.

Action:

Correct the invalid identifier.

More information:

Syntax Error Messages (see page 275)

ERPT254E

INVALID DATE FORMAT OPTION

Reason:

An invalid date format was detected.

Action:

Correct the format of the DATEFRMT operand of the OPTION command so that it is either mm/dd/yy or dd/mm/yy.

ERPT255E

FLASHBACK LIST MUST RUN AS SINGLE REPORT

Reason:

You have included more than one report with a flashback report.

Action:

Run each flashback report by itself.

ERPT257I

MAXIMUM RECORD COUNT EXCEEDED, EOF FORCED

Reason:

The number of input records read within the FROM/TO range exceeds the maximum value you set in the COUNT command. The reports you requested were printed, but the records exceeding the maximum value were ignored.

Action:

None.

ERPT258I

RECORDS READ BY REPORT WRITER *nnnnnn*

Reason:

The Report Writer read a total of *nnnnnn* input records.

Action:

None.

ERPT259I

RECORDS WITH TO/FROM RANGE *nnnnnn*

Reason:

nnnnnn

Provides the total number of input records read that passed the minimum FROM and maximum TO date and time criteria.

Action:

None.

ERPT260I

RECORDS PASSING SELECTION CRITERIA *nnnnnn*

Reason:

nnnnnn

Provides the total number of input records read that passed all selection and range criteria.

Action:

None.

ERPT261E

ERROR LOADING COMMAND TABLE *table* RC=*code*

Reason:

An attempt to load the command table identified by table did not complete successfully. Processing of reports terminates immediately.

code

Provides the return code of the load request.

Action:

Record the return code and then take appropriate action. If you need help, contact Technical Support.

ERPT262E

EOF WITH MISSING RUN COMMAND

Reason:

End-of-file was reached for SYSIPT, and no RUN command was encountered.

Action:

Include the RUN command and resubmit the report.

ERPT263E

RUN COMMAND ASSUMED

Reason:

The Report Writer encountered one or more END commands, but did not find a following RUN command. The Report Writer terminates.

Action:

Add a RUN command after the last END command and resubmit the job.

ERPT264E

ERRORS DETECTED IN *cname*

Reason:

An error was detected in an internal program name.

cname

Provides the name of a CSECT.

Action:

Contact Technical Support.

ERPT265E

TOO FEW COLUMNS AVAILABLE

Reason:

The EACH and PERIOD commands were used with the VPLOT command. The quotient of the PERIOD value divided by the EACH value must be less than or equal to the number of available plot columns. The Report Writer found that the resulting VPLOT report does not meet this criterion.

Action:

Increase the EACH value, decrease the PERIOD value, or alter both values. Alter the values so that the quotient of these values is less than or equal to the number of columns your report has available. The default number of columns is 110.

ERPT266E

INVALID LOG RECORD TYPE: type1 - Xtype2

Reason:

The Report Writer detected an invalid record type. type1 is the product type and type2 is the hexadecimal record type.

Action:

See whether you have an invalid or corrupt data file.

ERPT268I

REPORT# *nn* STORAGE ALLOCATED = *mmmmm*K BYTES

Reason:

The Report Writer allocated *mmmmm*K bytes of storage to report number *nn*.

The cause could be either of the following reasons:

- You asked to see what storage was allocated to a report by specifying the command OPTION(STORSTAT=YES).
- One or more reports have depleted your storage so that the Report Writer subsystem cannot run.

If your storage is depleted, one of these messages appears for every report you tried to run, following message ERPT201, MAIN STORAGE EXHAUSTED.

Action:

None.

ERPT269I

MOUNT TAPE FILE - *filename*

Reason:

The Report Writer finished reading input from the previous tape file and is asking you to mount the tape containing the next input file.

filename

Provides the name of the next input file.

The Report Writer requests the file names in the same order that you specified them with the INPUT command.

Action:

No action is required.

ERPT271E

ERROR LOADING MESSAGE TABLE *table* RC=*code*

Reason:

An attempt to load the message table identified by table did not complete successfully. Processing of reports terminates immediately.

code

Provides the return code of the load request.

Action:

Record the return code and then take appropriate action. If you need help, contact Technical Support.

ERPT272W

ERROR LOADING USER EXIT *name* RC=*code*

Reason:

An attempt to load the user exit identified by name did not complete successfully.

code

Provides the return code of the load request.

Action:

Record the return code and then take appropriate action. If you need help, contact Technical Support.

ERPT273I

VIRTUAL STORAGE AVAILABLE - *nnnn*K

Reason:

*nnnn*K of virtual storage is available at the end of the reporting process. This message is displayed if you specify the command OPTION(STORSTAT=YES).

Action:

None.

ERPT274I

STATUS: CNT - *nnnn* TIME - mm/dd/yy hh:mm:ss

Reason:

You asked to see the report status by specifying the command OPTION(RPTSTAT=YES).

nnnn

Provides the current input record count. The current input record is date-stamped mm/dd/yy and time-stamped hh:mm:ss.

Action:

No action is required.

ERPT275E

ERROR LOADING MODULE *module* RC=*code*

Reason:

An attempt to load the module identified by module did not complete successfully. Processing of reports terminates immediately.

code

Provides the return code of the load request.

Action:

Record the return code and then take appropriate action. If you need help, contact Technical Support.

ERPT276E

INVALID UTILITY REQUESTED - *name*

Reason:

The Report Writer requested an unknown or invalid internal utility.

name

Provides the name of the utility requested.

Action:

Contact Technical Support.

ERPT277E

FLASHBACK CONTROL RECORD NOT FOUND IN - EXPxxxx

Reason:

The Report Writer attempted to read the flashback file (EXPxxxx) but failed because it could not locate the flashback control record.

Action:

Change the DLBL statement to point to a valid flashback file.

ERPT278I

nnnnnnnn filename RECORDS

Reason:

The Report Writer successfully read nnnnnnnn physical records from the file identified by filename.

Action:

No action is required.

ERPT280E

INCOMPATIBLE VARIABLE TYPE

Reason:

This message identifies a syntax error. You attempted to use a character type variable in a report other than a TAB, TAB2, or FLASHBACK report. The variable above the dollar sign ($) is invalid.

Action:

Change the invalid variable or specify a TAB, TAB2, or FLASHBACK report.

More information:

Syntax Error Messages (see page 275)

ERPT281W

RECORD CORRUPTION DETECTED, RECORD IGNORED: Xtype VAR=variable

Reason:

The Report Writer has detected a corrupt record. Type is the hexadecimal record type, and variable is the internal name of the variable that detected the corruption.

Action:

Specify the command OPTION(DUMP=CORRUPT) and rerun the report. All corrupt records are now printed in dump format. Then contact Technical Support.

ERPT282I

DCMPGS OUTPUT CREATED: LIBRARY-library.sublibrary MEMBER - *member*

Reason:

Report Writer created the output formatted for use with RView in the member identified by member in the library identified by library.sublibrary.

Action:

None.

ERPT286E

INVALID EXPRESSION

Reason:

You used the DEFINE command with the VAR operand to define a new variable, but it contains an invalid arithmetic expression.

Action:

Correct the invalid expression and rerun the report. Delimit all components, operators, and parentheses by blanks.

ERPT287E

DUPLICATE VARIABLE NAME - CHOOSE DIFFERENT NAME

Reason:

You used the DEFINE command with the VAR operand to define a new variable, but the name you chose for the new variable exists.

Action:

Provide a different variable name and rerun the report.

ERPT289I

RECORDS DECOMPRESSED: *numrecs* COMPRESSION FACTOR: factor %

Reason:

The Report Writer has read a series of compressed records.

numrecs

The number of records decompressed

factor

The compression factor.

Action:

None.

ERPT290E

INCOMPATIBLE OPERANDS SPECIFIED

Reason:

This message identifies a syntax error. You have specified two or more incompatible operands. The operand above the dollar sign ($) is incompatible with a prior operand.

Action:

Correct the error and resubmit the report.

More information:

Syntax Error Messages (see page 275)

Appendix A: User Exits

This section contains the following topics:

About User Exits (see page 301)
Supplied User Exits (see page 301)

About User Exits

The Report Writer lets you specify a user exit routine. The Report Writer loads the exit routine during initialization and then passes control to the routine according to the exit command you have specified.

Use the DEFEREXT operand of OPTION command to defer the use exit call. Defer the call until after applying all selection criteria. Then have the Report Writer pass to the user exit only those records passing all selection criteria.

More information:

Commands (see page 39)

Supplied User Exits

Two user exit routines, EXPGUX and PRTGUX, are supplied with CA SYSVIEW. The source code for these exits is in the sample library. The following table shows the user exits, when the Report Writer passes control to each, and the exit command used to specify that routine.

User Exit	When Control Is Passed	Exit Command
EXPGUX	After each record is read and before the Report Writer processes the record	EXITRTN
PRTGUX	Before each line is printed	PRTEXIT

More information:

Commands (see page 39)

Source Code

The source code for the supplied user exits is in the EXPGUX and PRTGUX members of the *sysview*.SAMPLIB.

Important! If you modify either of the supplied user exits, make a backup copy of the member using another name. When you install a new release of CA SYSVIEW, PRTGUX and EXPGUX are overwritten. Having a backup copy of the member keeps your modifications from being lost.

Appendix B: Sample Reports

This section contains the following topics:

List of Sample Reports (see page 303)

List of Sample Reports

The Report Writer includes sample reports on many frequently requested topics. You can find supplied sample reports in the SAMPLIB.

The following sections list the available sample reports and give a brief description of each.

Important! The canned reports naming convention is not supported. You can find the associated member names for old canned reports in the SAMPLIB member $$INDEX.

CICS Sample Reports

The following lists the available CICS sample reports:

CICS.CICSSTAT

Member: CICS001

Displays CICS statistics broken down by individual CICS systems.

CICS.DCT

Member: CICS002

Displays a graph of transient data set usage.

CICS.MAXTIME

Member: CICS003

Displays a detailed tabular report all active transactions with a complete breakdown of all transaction maximum lifetime components.

CICS.OPERSTAT

Member: CICS004

Displays a tabular report of the transactions used by an operator and the performance that was achieved.

CICS.PROGUSE

Member: CICS005

Displays a graph of program usage.

CICS.TEMPSTOR

Member: CICS006

Displays a tabular report of all active transactions with various temporary storage statistics.

CICS.TERMIO

Member: CICS007

Displays a graph of terminal I/O for each terminal that has been used.

CICS.TERMRESP

Member: CICS008

Displays a graph of terminal response time for each terminal in the system.

CICS.TERMRESP.RANGE

Member: CICS009

Displays transaction usage broken down into terminal response-time groups.

CICS.TERMSTAT

Member: CICS010

Displays a tabular report all terminals that were active and how each performed.

CICS.TERMTRAN

Member: CICS011

Displays a graph of terminal usage and all transaction for each terminal.

CICS.TIMESTAT

Member: CICS012

Displays a detailed tabular report of transactions with a complete breakdown of all transaction lifetime components.

CICS.TRANFIO

Member: CICS013

Displays a graph that breaks down the transactions by their use and describes data set usage for each transaction, for each data set.

CICS.TRANLIFE.RANGE

Member: CICS014

Displays transaction usage broken down into specific transaction lifetime groups.

CICS.TRANRESP

Member: CICS015

Displays transaction usage broken down by their lifetime and shows terminal response time for each terminal used by each transaction.

CICS.TRANSTAT

Member: CICS016

Displays a tabular report of when transactions were used, how each transaction performed, and what resources it used.

CICS.TRANTERM

Member: CICS017

Displays a tabular report of all active transactions and the associated terminals.

CICS.TRANTIO

Member: CICS018

Displays a graph of terminal I/Os for each terminal a transaction used and the GRAF transaction use for each transaction used.

CICS.TRANUSE

Member: CICS019

Displays a graph of transaction use in the system.

CICS.TRANWAIT

Member: CICS020

Displays a graph of transaction waits.

CICS.PROGLIST

Member: CICS021

Displays a tabular summary of transactions with the programs they are calling and with basic statistics.

CICS.TRANLIFE

Member: CICS022

Displays a table of particular times constituting transaction lifetime in the first report and transaction wait time in the second report

CICS.LIFEDIST

Member: CICS023

Displays a table showing statistics of transaction lifetime intervals. Number of transactions and a percentage is displayed for several intervals.

CICS.DSNDIST

Member: CICS024

Displays a table showing statistics of data set usage by transactions. Number of transactions and a percentage is displayed for several intervals.

CICS.DB2DIST

Member: CICS025

Displays a table showing statistics of DB2 calls by transactions. Number of transactions and a percentage is displayed for several intervals.

CICS.STORDIST

Member: CICS026

Displays a table showing statistics of storage used by transactions. Number of transactions and a percentage is displayed for several intervals.

IMS Sample Reports

The following are the available IMS sample reports:

IMS.IMS01

Member: IMS01

Displays transaction lifetime breakdown and usage in tabular format.

IMS.IMS02

Member: IMS02

Displays the following graphs:

- Transaction usage by program specification
- Transaction lifetime sorted by logical terminals

IMS.IMS03

Member: IMS03

Plots the transaction usage over time.

IMS.IMSSUM01

Member: IMSSUM01

Displays various region processing time statistics for each program specification block.

IMS.IMSSUM02

Member: IMSSUM02

Counts DL/I calls to message queues for each jobname.

IMS.IMSSUM03

Member: IMSSUM03

Plots the summary of database DL/I calls.

IMS.IMSSUM04

Member: IMSSUM04

Shows the database operation counts for each user ID in tabular format.

MQ Sample Reports

The following are the available MQ sample reports:

MQ.REQLIST

Member: MQ001

Displays statistics on job activity sorted by time and summarized for all or selected objects.

MQ.REQFAIL

Member: MQ002

Displays statistics on job activity sorted by time and summarized for all or selected objects.

MQ.REQTIME

Member: MQ003

Displays request time statistics for jobs sorted by time and summarized for all or selected objects.

MVS Sample Reports

The following are the available MVS sample reports. You can produce MVS sample reports only if you run either CMF or RMF.

MVS.ASCB.STAT

Member: MVS001

Displays information about address space performance in tabular format.

MVS.CPU.PLOT.STAT

Member: MVS003

Displays a horizontal plot of CPU usage in one-hour intervals.

MVS.CPU.RESIDENCY.REPORT

Member: MVS004

Displays a scatter plot of job residency on the vertical axis and job CPU on the horizontal axis.

MVS.CPU.VPLOT.STAT

Member: MVS005

Displays a vertical plot of CPU use in 15-minute intervals.

MVS.DEVICE.STAT

Member: MVS006

Displays a tabular report of information about all active z/OS devices.

MVS.DISK.DSN.STAT

Member: MVS007

Displays a tabular report of exception counts on the ten most active data sets on each volume.

MVS.DISK.STAT

Member: MVS008

Displays a tabular report of information about all active z/OS DASD.

MVS.DSN.GRAF.STAT

Member: MVS009

Displays a graph of exception counts for active non-VSAM input and output data sets.

MVS.DSN.STAT

Member: MVS010

Displays a tabular report of exception counts for active non-VSAM input and output data sets.

MVS.PAGING.STAT

Member: MVS011

Displays a tabular report of various paging statistics.

MVS.SWAP.STAT

Member: MVS012

Displays a tabular report of various swap statistics by swap reason.

MVS.SYSTEM.STAT

Member: MVS013

Displays a tabular report of general z/OS system statistics.

MVS.VSAM.STAT

Member: MVS014

Displays a tabular report of various information for all active z/OS VSAM data sets.

AUDITLOG Sample Report

Member: AUDIT01

Displays a flashback report of AUDITLOG records.

COMMON Sample reports

Member: STEXC01

Displays a tabular report of state changes for State Exception records.

Index

*

* (asterisk)
  comment line delimiter • 19
  generic character in identifiers • 186
  in output report • 158

+

+ (plus sign) generic character in identifiers • 186

A

ALONE operand
  of PLOT command • 94, 228
  of PLOT2 command • 94
  of RANGE command • 95
ASCB variable class, MVS • 140
asterisk (*)
  comment line delimiter • 19
  generic character in identifiers • 186
  in output report • 158
AVERAGE line, printing in reports • 84
AVERAGE operand of OPTION command • 84
AVERAGED operand of EACH command • 54

B

blank lines
  in jobstream • 19
  including in reports • 84
blank spaces
  in commands • 19
  replacing zero values in reports • 84
BOTTOM operand
  of TAB command • 207
  of TAB2 command • 222
braces, convention for use of • 16
brackets, convention for use of • 16

C

CARDLEN command • 41
CDSA variable class, CICS • 113
CHANNEL variable class, MVS • 141
CHAR1 operand
  of GRAF command • 249, 250
  of GRAF2 command • 256, 257, 259
  of MPLOT command • 239, 246
  of PLOT command • 230
  of PLOT2 command • 233
  of VPLOT command • 236
  of VPLOT2 command • 238
CHAR2 operand
  of PLOT command • 230
  of PLOT2 command • 233
charge-back reports
  COST command • 42
  examples • 219
CICS variables • 112, 140
column headings
  changing in tabular reports • 198
  meanings of in FLASHBACK LIST reports • 262
comma, for continuing to new line • 19
commands
  blank lines in jobstream • 19
  blank spaces in jobstream • 19
  CARDLEN • 41
  case • 19
  COMMENT • 41
  comment lines in jobstream • 19
  continuing to new line • 19
  COST • 42
  COUNT • 44
  DEFINE • 45
  EACH • 54
  END • 27, 56
  EXCLUDE • 57
  EXITRTN • 58
  FLASHBACK AUDIT • 59
  FLASHBACK LIST • 60
  FROM • 62
  GLOBAL • 66
  GOTO • 67
  GRAF • 247
  GRAF2 • 70
  history reporting • 19
  IF • 77
  INCLUDE • 78
  INVOICE • 80
  PERIOD • 91
  PLOT • 92
  PLOT2 • 94

position in line • 19
processing order • 19
PRTEXIT • 94
RANGE • 95
RUN • 27, 97
SELECT • 97
SHIFT • 98
TABulate • 100
TITLE1 • 104
TITLE2 • 104
TITLE3 • 104
TO • 62
used with variables • 40
variables, ID options • 40
VPLOT • 106
COMMENT command • 41
comment line
asterisk delimiter(*) as • 19
in output reports • 41
multiple comment lines • 19
continuation character in commands • 19
COST command
description • 42
example • 42, 219
related options • 42
related variables • 42
COUNT command • 269, 270
COUNT variable option • 158
CPU variable class, MVS • 142
create a job stream for a report • 25
CSVGEN Utility • 35

D

DATASET variable class, CICS • 113
dates
excluding dates or date ranges • 57
for limiting report data • 63, 64
format in reports • 84
include in a tabular report • 193
including dates or date ranges • 78
including in report titles • 104
DATETIME operand of the TAB command • 193
DAY operand, of the TAB command • 194
decimal positions in output fields • 158
default value, convention for presenting • 16
DEFERRED operand
of COST command • 42
of RANGE command • 95, 212, 271
DEFINE command
defining new variables • 50
examples • 50
DEVICE variable class, MVS • 143
diagnostic dump of input • 84
diagnostic tracing facility • 84
DIFF variable option • 201
disk space, minimizing use of • 84
DISK variable class, MVS • 143
DLI variable class, CICS • 114
DSN variable class, MVS • 144
DUMP operand of OPTION command • 84
dump, printing diagnostic of input • 84

E

EACH command
description • 54
with GRAF command • 248, 253
with HPLOT and VPLOT • 244, 246
with HPLOT command • 241
with MPLOT command • 239
with PLOT command • 226
with PLOT2 command • 231
with TAB command • 193, 198
with VPLOT command • 234, 235
ECDSA variable class, CICS • 116
ellipsis, convention for use of • 16
END command • 27, 56
ending a report • 56
end-of-file, when maximum TO time is exceeded • 84
ERDSA variable class, CICS • 116
ERPT267 message • 84
ERPT274 message • 84
errors, excluding messages from print and display • 84
EUDSA variable class, CICS • 117
EXCEPTION variable class, CICS • 117
EXCLUDE command • 57
exclude identifiers with generic character • 186
EXCP variable class, MVS • 145
EXITRTN command • 58, 301
EXPGUX (supplied user exit) description • 301

F

first day of the week, specifying • 84
FLASHBACK AUDITLOG command • 59
FLASHBACK LIST command • 60

default report • 262
generate file type of CSV • 34
limiting output • 262
report description • 262
restriction for running reports • 261
FOR operand
of GRAF command • 249, 250, 251
of GRAF2 command • 256
of TAB command • 205
of TAB2 command • 220
FORMAT variable option • 158
formatting identifiers • 187
FROM command • 62
with FLASHBACK LIST • 262, 270
with GRAF command • 247
with MPLOT command • 239
with PLOT command • 226
with TAB command • 193
with VPLOT command • 234, 235

G

generic characters in identifiers
descriptions • 186
in GRAF report • 248
in PLOT report • 227
in PLOT2 report • 231
in TAB report • 205, 212, 216, 219
GLOBAL command • 66
global commands
definition of • 14
list of • 27
global value, overriding • 66
GOTO command • 67
GRAF command • 247
GRAF2 command • 70
graph reports
changing the scale of • 250, 257
defined • 247
sorting • 251
GROUP command • 72, 215

H

HDR1 operand of TAB command • 197, 198, 219
HDR2 operand f TAB command • 198
HDR2 operand of TAB command • 197, 198
HEADER command • 74
hexadecimal, specifying identifiers using • 187
HIGHLIGHT operand of RANGE command • 95
historical file
JCL for reports using • 25
using as input • 25
history reporting
commands • 19
commands, writing • 27
concepts and terminology • 14
memory requirements • 14
purpose • 13
HPLOT command
with MPLOT command • 244, 246
with VPLOT command • 241

I

ID operand of EXCLUDE command • 57
ID options
definition of • 14
introduction to • 24
specifying identifiers with • 186, 187
using with commands • 177
ID Options and Identifiers, using • 24
identifiers
assigning new values for • 77
definition of • 14
formatting • 187
generic characters • 186
hexadecimal specification • 187
introduction to • 24
IF command • 77
INCLUDE command • 78
input
limiting records read, specifying date and time • 62
printing dump of used or corrupted records • 84
producing statistics on • 84
using both MVS and CICS data • 197
using records with negative values • 159
zero values, including records with • 163
INPUT command • 197
interval TRANSACTION variable class, CICS • 124
INVOICE command • 80
IO variable class, MVS • 146
italics, conventions for use in JCL • 16

J

JCL for running reports • 25
JOB variable class, MVS • 147

JOBNAME operand of FLASHBACK LIST command • 267

L

line count, setting for all reports • 82
line length setting • 41
LOGICAL variable option • 159
lowercase
  convention for use of • 16
  using in commands • 19
LPAR variable class, MVS • 148

M

MAX variable option • 159
MAXDATE variable option • 160
MAXTIME variable option • 160
MAXTO operand of OPTION command • 84
memory requirements • 14
messages
  ERPT267 • 84
  ERPT274 • 84
  excluding from print and display • 84
  status messages • 84
MIN variable option • 159
MPLOT command
  examples • 239
  with the HPLOT command • 244, 246

N

negative values, treating as unsigned • 159
not sign (?) generic character in identifiers • 186
NULLINE operand of OPTION command • 84
NXTIME variable option • 164
NXTND variable option • 165

O

operand, definition of • 14
OPERATOR variable class, MVS • 148
options
  functions • 109
  setting for all reports • 84
or-sign, convention for use of • 16

P

PAGE variable class, CICS • 118
PAGE variable class, MVS • 149
PAGESWAP variable class, MVS • 151
parentheses, convention for use of • 16
PERCENT variable option • 161
performance groups • 72
performance TRANSACTION variable class, CICS • 124
PERFORMANCE variable class, MVS • 151
PERIOD command • 91
PGMSTOR operand of OPTION command • 84
planning reports • 21
PLOT command • 92
plot reports, description of • 225
PLOT2 command • 94
plotting
  resource usage in vertical columns • 106
  two variables • 94
  two variables across time • 107
plus sign (+) generic character in identifiers • 186
primary report commands
  definition of • 14, 28
  list of • 29
PRINT variable class, MVS • 152
printing variables
  as a maximum or minimum value • 159
  as a total • 163
  date of highest value • 160
  fewer digits • 165
  more digits • 165
  time of highest value • 160
  zero values • 84
PROCESSOR variable class, MVS • 153
PROGram variable class, CICS • 119
PRTEXIT command • 94, 301
PRTGUX, supplied user exit • 94, 301

R

RANGE command
  description • 95
  examples • 216
  with FLASHBACK LIST command • 271
  with GRAF • 254
  with TAB command • 212
RANGE operand of OPTION command • 84
RANGE variable option • 215
RANGE, line in plot reports • 84
RATE variable option
  description • 161
  with PLOT • 231
  with PLOT2 • 233

with TAB command • 197, 204
RECAP operand of OPTION command • 84
RECSTAT operand of OPTION command • 84
REGID operand of FLASHBACK LIST command • 267
region size, recommended • 14
REGION variable class, CICS • 119
report
create a job stream • 25
report commands, definition of • 14
Report Writer
commands, definition of • 14
concepts • 14
introduction to • 13
memory requirements • 14
terminology • 14
users • 13
reports
creating file type of .csv • 34, 37
defining field length • 158
formats available • 22
including lines with no activity in reports • 84
information you want to include • 23
input, limiting records read • 63
JCL, using historical file • 25
minimal commands for writing • 31
multiple • 32
overview of • 13
planning • 21
printing zeros in • 84
purpose • 13
tailoring • 24
tailoring with secondary report commands • 33
titles • 104
types of • 22
RIGHT variable option • 74
RPTSTAT operand of OPTION command • 84
RUN command • 97

S

sample reports
CICS • 303
IMS • 307
list of • 303
MQ • 308
MVS • 309
SCALE operand
of GRAF command • 250, 251
of GRAF2 command • 257, 258
of HPLOT command • 241, 243
of MPLOT command • 239
of PLOT command • 229
of VPLOT command • 235, 236
of VPLOT2 command • 238
secondary report commands
definition of • 14, 30
using to tailor reports • 33
SELECT
operand of COST command • 42
variable option • 202
SELECT command • 97
SELECTED operand of COUNT command • 44, 270
SELZERO variable option • 163
SHIFT command • 252, 253
SORT operand
of GRAF command • 251
of TAB command • 210
sorting
graph reports • 251
tabular reports • 209
STACK operand of VPLOT command • 237
statistics on input data records • 84
STEP variable class, MVS • 147
storage
acquiring above or below 16M line • 84
operand of OPTION command • 84
printing storage statistics • 84
tracing storage requests • 84
used by report writer, displaying • 84
variable class, MVS • 154
STORSTAT operand of OPTION command • 84
SU (service units) variable class, MVS • 155
summary line in tabular reports • 84
SUMMARY operand of OPTION command • 84
SWAP variable class, MVS • 155
syntax conventions
applying • 18
list of • 16
SYSTEM variable class, CICS • 120

T

TAB2SKIP operand of OPTION command • 84
tabular reports
introduction to • 191
producing • 100
sorting • 209
TABulate command • 100

TEMPSTOR variable class, CICS • 120
TERMINAL variable class, CICS • 121
THRESHOLD variable class, CICS • 122
time
  defining shifts • 98
  defining time periods • 91
  excluding time ranges • 57
  include column for in tabular report • 193
  including column for in tabular report • 193
  including in report titles • 104
  including time ranges • 78
  period for report • 54
  specifying for report data • 63, 64
TIME operand
  of FLASHBACK LIST command • 269
  of TAB command • 193, 196
TITLE commands, keywords • 104
title in a tabular report • 74
TITLEn command • 104
TO command
  description • 62
  maximum TO time checking • 84
  with FLASHBACK LIST • 262, 270
  with GRAF command • 247
  with MPLOT command • 239
  with PLOT command • 226
  with TAB command • 193
  with VPLOT command • 234, 235
TOP operand
  of TAB command • 207
  of TAB2 command • 222
TOTAL variable option • 163
trace (diagnostic tracing facility) • 84
TRACE operand of OPTION command • 84
transactions, reports on • 60
TRANSIENT variable class, CICS • 128
TSO variable class, MVS • 156

U

UDSA variable class, CICS • 129
UNDERLINE operand of RANGE command • 95
underscoring, convention for use of • 16
uppercase
  convention for use of • 16
  using in commands • 19
user exits
  processing order • 301
  supplied user exits • 301
user exits, defining exit to be called
  before each line is printed • 94
  EXITRTN command • 301
  PRTEXIT command • 301

V

variable names, structure and parts • 110
variable options
  COUNT • 158
  definition of • 14
  DIFF • 201
  FORMAT • 158
  LOGICAL • 159
  MAX • 159
  MAXDATE • 160
  MAXTIME • 160
  MIN • 159
  NXTIME • 164
  NXTND • 165
  PERCENT • 161
  RATE • 161
  SELECT • 202
  SELZERO • 163
  TOTAL • 163
  using • 23
  XTIME • 164
  XTND • 165
variables
  changing from count to rate • 204
  convention for presenting • 16
  definition of • 14, 109
  digits in output • 165
  forming • 110
  negative values in input • 159
  overview • 23
  parts of • 110
  plotting the distribution of one, based on the value of another • 76
  purpose of • 23
  SMF field names • 112
  source for MVS • 112
  used with commands • 40
  zero values, including records with • 163
VARS operand of GRAF2 command • 258
vertical bar, convention for use of • 16
vertical plot, producing • 83
virtual storage, required • 14
VPLOT command • 106

VPLOT command with the HPLOT command • 241

W

WEEK operand of OPTION command • 84
writing multiple reports • 32

X

XMSG operand of OPTION command • 84
XTIME variable option • 164
XTND variable option • 165

Z

zeroes
    in input, including records with • 163
    including lines with no activity in reports • 84
    printing in reports • 84
ZEROFLD operand of OPTION command • 84