

CA Deliver™

Reference Guide

Release 12.1



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 © 2013 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 11™
- CA ACF2™
- CA Balancing
- CA Common Services
- CA Dispatch™
- CA Output Management Document Viewer
- CA Output Management Web Viewer
- CA Roscoe®
- CA Spool™
- CA SYSVIEW
- CA Top Secret
- CA View™

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Contents

Chapter 1: Introduction	15
Who Should Read This Guide?	15
New Features and Enhancements in this Release.....	15
ServiceDesk Interface.....	16
Health Checker	16
Operating System and Release Level Prerequisites	16
Working with CA View.....	17
Illustration of Tasks	18
How the Started Task Works.....	19
How Tasks Work: Pre-Spool	19
How Tasks Work: Post-Spool	21
How to Use Network Input with a Shared Database	23
How to Use CA Deliver in a Multi-Sysplex.....	24
How the Started Task Data Sets Are Allocated	25
Remote Mirroring	26
Using a Test Database	27
Using the SYSID Initialization Parameter.....	27
Types of Report Processing	27
Basic Report Processing	28
Stacked Report Processing.....	28
Interleaved Report Processing	29
Monitored Data Output	30
Control Break Report Processing	30
Obtaining Print Attributes.....	30
Print Attributes	31
SMF Record Layouts	33
 Chapter 2: Initialization Parameters	 35
Overview of Initialization Parameters.....	35
Initialization Parameter Syntax	35
Syntax Rules	36
Inserting Initialization Parameter Comments	36
JOBCLSL Initialization Parameter	36
BNDLMOUT, Attributes, and the First Report in a Bundle	37
Bundle Output Task and the BOT Initialization Parameter	37
Initialization Parameters	38

ARCH	38
ARCHnn(Form 1)	39
ARCHnn(Form 2)	40
AUTHTID	42
AUTOACT	42
BANNER	43
BEGINDAY	43
BNDLBNR1	43
BNDLBNR2	44
BNDLBNR3	44
BNDLCLS	45
BNDLCONF	45
BNDLDEST	46
BNDLHDTL	46
BNDLINT	47
BNDLMOUT	47
BNDLSCAN	48
BNDLWAIT	48
BOT	48
CCX	49
DAYS	50
DEFDEL	50
DEFOUT	51
DEFREPT	52
DEFRVIEW	53
DYNRCHAR	54
EFORMAT	55
EMAILQ	55
EXTSEC	56
EXTTRAN	56
FEATURE	57
FREEALL	59
HDETAIL	60
HISTCNT	60
INBSSN	61
JOBCLSL	61
JOBREF	62
LOGO	62
MAXHIST	63
MAXJESQ	63
NAME	63
NETCLSL	64

NETDEST	64
NETFORM	65
NETONLY	65
NETREQUE	66
NETRERUN	66
NETUNDEF	67
OFFPW	67
OUTPUT	67
PASSTHRU	68
PRBTASK	68
PREVRUN	69
REDIS	69
RMSWARN	70
RPTENQ	70
RPTNPROD	71
SAR	71
SARBUFACT	72
SECMSG	72
SETCMD	72
SETPAGE	73
SETPW	73
SMF	74
SMF30	74
START	74
STKCHNn	75
STKMODE	75
STNAMEn	76
STKNOTXT	76
STOPPW	76
SYSCLSL	77
SYSID	77
TEXT	78
TIME	78
WARNING	79
WEBSVRn	79
WRITER	79

Chapter 3: Operator Commands 81

Starting	82
Refreshing System Routines	82
Stopping	83

Withdrawing.....	83
Freeing Reports and Bundles from a System	84
Starting and Stopping CA Balancing Report Control	84
Starting the Bundle Output Task	84
Stopping the Bundle Output Task	84
Posting the Bundle Output Task.....	85
Resetting the In-Storage Buffers	85
Displaying Currently Active Parameters.....	85
Setting an Initialization Parameter While the Product Is Running	86
Initialization Parameter Considerations.....	86
Unlocking the Checkpoint	87
Operating the Cross-Memory Task	87
Starting the Cross-Memory Task (START)	88
Immediate Termination (SHUTDOWN)	88
Normal Termination (QUIESCE)	88
Canceling Users (CANCEL)	89
Suspending Additional Logons (SUSPEND).....	90
Resuming Acceptance of New Logons (RESUME)	90
Closing Cross-Memory Tasks (CLOSE)	90
Reopening Cross-Memory Tasks (OPEN)	91
Listing Statistics (LIST)	91
Changing Cross-Memory Parameters Online (CHANGE).....	92
Defining Multiple VTAM Cross-Memory Regions.....	92
Regions Grouped by Operating System Subsystem Name.....	93
Regions Separated by REGIONIDs and APPLID.....	94
Checking the Status of Multiple Regions	94
Specifying the Parameters: SYSIN Statements.....	94

Chapter 4: Utilities 97

What Is a Control Statement?	97
Example.....	97
Syntax Rules	98
Saving Time with Conversion Utilities RMODBB and RMOJCL	98
How to Quickly Set Up a Database.....	98
RMOBPR - Batch Bundle Posting	99
Example.....	99
Job Control Statements.....	100
RMOCPMAP - Checkpoint Map Utility	100
Sample JCL.....	101
Checkpoint Space Analysis	102
Checkpoint Contents Report	103

RMODBASE - Database Maintenance	104
Job Control Statements.....	105
ADDDDS Control Statement	106
BLOAD Control Statement.....	109
CONVERT Control Statement	111
COPY Control Statement	112
DELBAN Control Statement.....	113
DELETE Control Statement	114
DELPAN Control Statement.....	114
DELPRSET Control Statement.....	114
HDELETE Control Statement	115
LOAD Control Statement.....	115
MAKECKPT Control Statement	117
NAME Control Statement.....	119
OLOAD Control Statement	119
PLOAD Control Statement.....	120
RENAME Control Statement	120
STATUS Control Statement	121
UNLOAD Control Statement.....	123
VERSION Control Statement	124
Examples of Control Statements.....	124
RMODBB - Database Construction from Existing Data	127
Updating Active Definitions	128
Job Control Statements.....	129
DBASE Control Statement	129
Data Definition Control Statements.....	130
BNDLDEF Control Statement.....	131
DISTDEF Control Statement	135
JOBDEF Control Statement	140
RPTDEF Control Statement	143
Using Combinations of Parameters and Functions	156
RMODSC Report Data Set Collector	163
Data Set Collector Considerations	163
Job Control Statements.....	164
NAME Control Statement.....	165
COLLECT Control Statement.....	165
Input User Exit.....	166
RMOGRW - General Report Writer	167
Job Control Statements.....	168
Control Statements	169
Field Names.....	170
Sequence in Which Database Records Are Accessed.....	171

Database Fields for Job Descriptor Record	172
Database Fields for Report Descriptor Record	173
Database Fields for Report History Record	179
Database Fields for Report Detail History Record	181
Database Fields for Distribution Data Record	182
Database Fields for Bundle Descriptor Record	185
Database Fields for Bundle History Record	186
Database Fields for Bundle Detail History Record	187
Database Fields for Active Report Status Record	188
Database Fields for Active Bundle Status Record	189
Reserved Fields	190
Expression	191
Condition	195
BREAK Control Statement	196
CONTINUE Control Statement	196
CONTROL Control Statement	197
DEFINE Control Statement	199
DO Control Statement	201
ELSE Control Statement	203
END Control Statement	203
IF Control Statement	204
NEXT Control Statement	205
ON Control Statement	205
OUTPUT Control Statement	206
PRINT Control Statement	208
RELEASE Control Statement	210
SELECT Control Statement	210
SET Control Statement	212
SORT Control Statement	213
STOP Control Statement	214
THEN Control Statement	215
TITLE Control Statement	215
RMOHTP - Batch Detail History Reporting	225
Job Control Statements	226
DATA Control Statement	226
DATE Control Statement	227
NAME Control Statement	227
TIME Control Statement	227
Station Data	228
RMOIFMAP - Index File Mapping Utility	229
Sample JCL	229
RMOJCL - Automatic Database Construction From JCL	229

Syntax.....	230
Job Control Statements.....	231
EXCLUDE Control Statement	232
NAME Control Statement.....	232
OMIT Control Statement.....	232
SELECT Control Statement	234
RMOJCS - Enhanced Database Construction from JCL	234
Syntax.....	235
Job Control Statements.....	237
EXCLUDE Control Statement	238
NAME Control Statement.....	238
OMIT Control Statement.....	238
SELECT Control Statement	239
RMOPRE - Take Action on the Most Recently Produced Reports	239
Syntax.....	239
Using AFP ACIF to Archive Reports	240
Job Control Statements.....	240
NAME Control Statement.....	240
JOBNAME Control Statement	240
JOBNUM Control Statement	241
FROM-STEP Control Statement.....	241
TO-STEP Control Statement	241
EXCLUDE Control Statement	242
LSERV-INBSSN Control Statement.....	242
Using Page and Form Definitions	243
Page and Form Definition Options	243
How RMOPSF Works	244
\$BANNER and \$\$FIRST	244
Using RMOPSF to Correct Banner Page Problems	244
RMOPSF - Form and Page Definition Modification	246
Contents of FORMDEF and PAGEDEF.....	246
Job Control Statements.....	248
MDLFDEF and MDLPDEF Control Statements	249
REMOVE Control Statement	249
EXCLUDE Control Statement	249
SELECT Control Statement	249
RMORAP - Activating and Deactivating Reports from Batch.....	250
Job Control Statements.....	252
JOB Control Statement.....	253
NAME Control Statement.....	253
REPORT Control Statement	253
TYPE Control Statement	253

RMORMS - Using Tracking.....	255
Syntax.....	255
Using AFP ACIF to Archive Reports	255
Job Control Statements.....	256
NAME Control Statement.....	256
FROM-NODE Control Statement	256
LSERV-INBSSN Control Statement.....	257
Rerun Processing Status Report.....	257
RMORPT - Batch Reporting	258
Job Control Statements.....	259
NAME Control Statement.....	260
REPORT Control Statement.....	260
RMORXB - Rebuilding Cross-Reference Records	263
Job Control Statements.....	263
RMOUTIL - Migration Support.....	264
Rule for Updating Active Definitions.....	264
Job Control Statements.....	265
COPY Control Statement	265
DELETE Control Statement	267
NAME Control Statement.....	268
RENAME Control Statement	269

Chapter 5: The Database 271

What is the Database?	271
Elements of the Database	271
Rules and Guidelines	271
Database Data Set Attributes.....	272
Types of Records in the Database	273
Structure of the Checkpoint Data Set	274
Checkpoint Data Set Attributes.....	275
Types of Data Maintained in the Database	275
Utilities for Building and Modifying the Database	276
Specifying Job Name Translation	277
Rules for Job Name Translation Control Statements	277

Chapter 6: Model Banner Pages 281

Model Banner Page Library	281
Using Model Banner Page Members.....	281
Example 1	282
Example 2	283
Example 3	284

Example 4	285
Types of Model Banner Page Members	286
Specifying a Banner Page	286
Use Attribute Characters	287
Converted Block Letters	288
Substituting Text of Varying Lengths	288
Use Control Statements for Model Banner Page Members	288
/BEGSEP Control Statement	288
/ENDSEP Control Statement	289
/ATTR Control Statement	289
Using Carriage Control Characters	290
Carriage Control on a Data Line	290
Types of Carriage Control Characters	290
Rules for Carriage Control Characters	291
Use Symbolic Variables	291
Example	291
Types of Symbolic Variables	292

Chapter 7: Setting Up Print Attributes 299

Format of Type 6 SMF Records	299
Example 1	299
Example 2	300
Printer Setup Member	303
Format of the Records in a Printer Setup Member	303
Loading Printer Setup Library Members to the Database	303
Appending Printer Setup Records to a Report	303
Tailor Banner Pages to Print on a Xerox 9700 Printer	304
Tailoring Model Banner Pages	304
Tailoring Bundle Model Banner Pages	305

Chapter 8: Accessing Programs from Global Subsystem 307

Establishing a Host Command Environment	307
Commands Processed by the Host Command	307
GREXX Variables	308
Example of a REXX EXEC	309

Chapter 9: NJE Unattended Download 311

Overview of Unattended Downloading	311
Software Requirements	311
Sending a Report or Bundle for Document Viewing	312

Chapter 10: Security	313
Internal Security	313
RMOATHTB Security.....	314
Required Access Levels	315
Using Macros to Define Authorization Criteria	316
RMOAGRP Macro	316
Authorization Group Criteria.....	318
RMOATH Macro	319
RMOAEND Macro.....	320
Coding Macros	321
External Security	326
Initialization Parameters	326
Resources and Authorizations	327
Implementing External Security for CA Top Secret.....	331
Implement External Security for CA ACF2.....	333
Implementing External Security for RACF	338
 Appendix A: CA Deliver Health Checks	 343
The Health Checks	343
DLVR_OPT_HDETAIL.....	343
DLVR_PRFM_PQE.....	344
Messages.....	345
 Index	 349

Chapter 1: Introduction

Welcome to CA Deliver, a document management product that provides automatic distribution of documents that were created on the mainframe.

This chapter explains how CA Deliver works and how it produces reports, and provides the other information you need to know to use CA Deliver.

This section contains the following topics:

[Who Should Read This Guide?](#) (see page 15)

[New Features and Enhancements in this Release](#) (see page 15)

[Operating System and Release Level Prerequisites](#) (see page 16)

[Working with CA View](#) (see page 17)

[Illustration of Tasks](#) (see page 18)

[How the Started Task Works](#) (see page 19)

[Using a Test Database](#) (see page 27)

[Types of Report Processing](#) (see page 27)

[Obtaining Print Attributes](#) (see page 30)

[SMF Record Layouts](#) (see page 33)

Who Should Read This Guide?

This guide is targeted to system administrators who are responsible for managing report distribution and tracking.

This guide assumes that:

- You are familiar with IBM computer system terms and concepts
- You have a basic working knowledge of IBM JCL
- You have a working knowledge of MVS online facilities because CA Deliver panels behave like ISPF panels

New Features and Enhancements in this Release

For information about:

- New features and enhancements, see the *Release Notes*
- The conventions used in this guide, see the *Administration Guide*
- User abend codes and routine error messages, see the *Message Guide*

ServiceDesk Interface

The CA Deliver ServiceDesk interface automatically opens service desk issues in the CA Service Plus Help Desk database.

CA Deliver uses CAISDI/med services to interface with the CAISDI/soap server task to open a service desk issue that is based on the occurrence of a CA Deliver abnormal termination.

Health Checker

The CA Deliver Health Checker feature raises alerts about conditions that could prevent CA Deliver from running properly, if left uncorrected.

Health Checker guides you to address the problem by providing best practice suggestions for running CA Deliver.

For more information, see the "CA Deliver Health Checker" appendix later in this document.

Operating System and Release Level Prerequisites

Be aware of the following:

- For CA View Output Archival and Viewing (CA View) to work properly, you must use CA View Release 11.5 or later with CA Deliver Release 12.1 .
- You must use an IBM Supported z/OS version.
- CA Deliver uses the dynamic allocation facilities to allocate IBM 3800 printer attributes such as BURST, CHARS, and MODIFY to print reports.
- You need CA GSS for RMODBASE, RMODBB, and RMOGRW GREXX communication.
CA GSS is included with CA Common Services.

Working with CA View

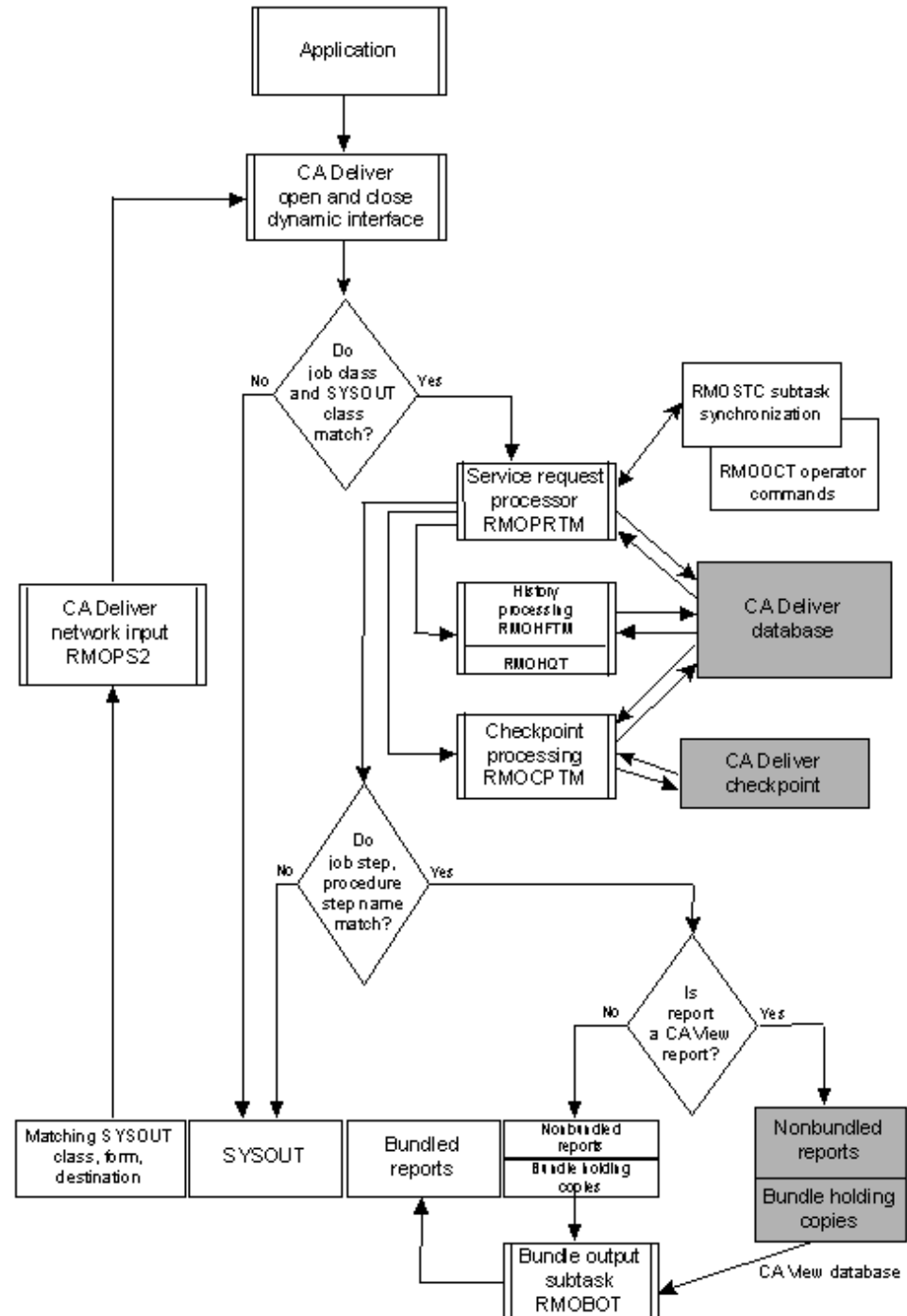
If you use CA Deliver with CA View, you can:

- Automatically distribute reports to disk
- Back up reports to tape
- View reports online
- Reprint reports

CA View is described in your CA View documentation.

Illustration of Tasks

This illustration shows how CA Deliver tasks work.



How the Started Task Works

The CA Deliver task is a started task that controls the production of reports and tracks the distribution of reports. It is automatically marked as "non-swappable."

This task sets the dynamic interface between CA Deliver and your operating system.

This section explains how:

- Tasks work: pre-spool
- Tasks work: post-spool
- To use network input with a shared database
- To use CA Deliver in a multi-sysplex environment
- The started task data sets are allocated

How Tasks Work: Pre-Spool

When you start the started task on each operating system at your site, each started task should remain active until the operating system on which it is running is brought down.

The started task obtains control whenever a data set is opened or closed by a job. The started task first checks to see whether the SYSOUT is a SYSOUT that it is to control.

Open Processing

For open processing, the started task verifies whether:

- The class of the job is specified by the JOBCLSL initialization parameter
- The SYSOUT class is specified by the SYSCLSL initialization parameter
- The job name or the alternate job name (optionally based on the JOBREF initialization parameter) is defined for the database
- The job-step name and procedure step name match
- One or more report definitions exist in the database for the DD statement that is being opened
- The name of the program being executed is not SARINIT, SARDBB, SARDBASE, RMODBB, or RMODBASE.

Note: The following CA View and CA Deliver utility programs SARINIT, SARDBB, SARDBASE, RMODBB, or RMODBASE output require post spool processing. Pre-spool capture of these CA View and CA Deliver programs' output is no longer allowed. Reports generated by these programs are not candidates for dynamic report creation.

Report Processing

Once the started task verifies that the SYSOUT being opened is to be controlled, it modifies the Access Method Control block for the data set so that it gains control every time a record is written for the data set. Basic reports are opened when the DD statement is opened. Stacked and interleaved reports are opened the first time that their selection criteria are matched.

Each record is written to every allocated and open report data set.

- CA Deliver selected the data set when it was opened
- Closed all open report data sets
- For not-produced reports, CA Deliver sets the NPROD flag ON and optionally writes a SYSOUT line based on the setting of the RPTNPROD initialization parameter

Close Processing

For close processing, the started task verifies that CA Deliver selected the data set when it was opened.

Note: If your application opens and closes the same SYSOUT data set more than once while executing, we recommend that you use post-spool processing to direct all output to the same report.

How Tasks Work: Post-Spool

Network input is a post-spool operation of CA Deliver whereby reports can be written to the JES spool first by an application, then later retrieved by CA Deliver for processing.

Network input was originally intended to enable a report created at a node in a multi-node network (that did not run CA Deliver) to be distributed and tracked by CA Deliver when it was routed for printing to another node in the network that *did* run CA Deliver.

However, the operation now allows any report that can be written to the JES spool regardless of its node in the network (and then later retrieved by CA Deliver) to be processed through network input.

For a report to be input to CA Deliver through network input, the report must be:

- Defined in the database
- Created with the same SYSOUT attributes that were defined to the task for network input at initialization time.

One or more of the following attributes can be used to define network input to CA Deliver:

- The SYSOUT class (specified by the NETCLSL initialization parameter)
- The SYSOUT destination (specified by the NETDEST initialization parameter)
- The Forms name (specified by the NETFORM initialization parameter)

CA Deliver uses the IBM SYSOUT Application Programming Interface (SAPI).

SYSOUT Application Interface

With the SYSOUT Application Interface, network input data sets are processed as follows:

- Network input data sets from a job or a DD statement that are not defined to database are created under an UNDEF report if the NETUNDEF initialization parameter is set to YES.

If an UNDEF report is not defined or the NETUNDEF initialization parameter is set to NO, the network input data is requeued to or copied to a new data set under the new class, new destination, and/or new forms referenced by the NETCLSL, NETDEST, and NETFORMS initialization parameters. The NETREQUE initialization parameter determines whether these network input data sets are requeued or copied.

- The copied, distributed, or UNDEF reports appear under the name of the started task.
 - Requeued network input data sets retain their original job name, job ID, and accounting information.
 - These network input data sets reside in a non-held status even if the new class references a held class.
 - If an error occurs while re-queuing the network input data set, the network input data set is placed in a held status.
- Output statement parameters, such as PAGEDEF, FORMDEF, NAME, ADDRESS, TITLE, and so on, are retrieved and processed the same as in the pre-spool environment.
- Accounting information is obtained.

Reports distributed through network input do not have the accounting information attached, but it is accessible through user exits and it is retained in archival copies of the report.
- The copies parameter is supported for report distribution specifications that indicate copies as "*".
- COPIES= supplied on a /*JOBPARM statement is not captured.

Example 1

Assume that CA Deliver runs at node N1 and that SYSOUT class E has been reserved and defined to CA Deliver for network input.

To have a report routed to and printed by CA Deliver, it must be created with the following attributes:

```
DEST=N1  
SYSOUT=E
```

From another CPU, for non-shared JES spool, the JCL is as follows:

```
//STEP1    EXEC  PGM=IEBGENER
//SYSUT1   DD    DSN=JCL.DATA(LISTING) ,DISP=SHR
//SYSUT2   DD    SYSOUT=E,DEST=N1
//SYSIN    DD    DUMMY
//SYSPRINT DD    SYSOUT=A
```

In this example, the initialization parameter NETCLSL must be set to NETCLSL=E/A, where the value A is the class to which SYSOUT data is to be written if the extracted job does not match the definition in the database.

Example 2

Assume that CA Deliver runs at node N1 and that destination R100 has been reserved and defined to CA Deliver for network input. To have a report routed to and printed by CA Deliver, it must be created with the following attribute:

```
DEST=N1R100
```

From another CPU, for non-shared JES spool, the JCL is:

```
//STEP1    EXEC  PGM=IEBGENER
//SYSUT1   DD    DSN=JCL.DATA(LISTING) ,DISP=SHR
//SYSUT2   DD    SYSOUT=B,DEST=N1R100
//SYSIN    DD    DUMMY
//SYSPRINT DD    SYSOUT=A
```

In this example, the initialization parameter NETDEST must be set to NETDEST=R100.

How to Use Network Input with a Shared Database

If you use network input and share one database across more than one CPU in a shared JES spool environment, you must set initialization parameters NETCLSL, NETDEST, and NETFORM to unique values on each CPU.

Although network input is an alternative to pre-spool operation of CA Deliver, it is not recommended as a substitute for pre-spool operation because:

- Network input necessitates additional overhead because the reports must be both written to and read from the JES spool before CA Deliver can process them.
- There is a delay when CA Deliver processes reports--they are not normally available to CA Deliver until after the job that creates them is completed.

How to Use CA Deliver in a Multi-Sysplex

CA Deliver can capture work from other sysplexes or system complexes. The network input feature provides this functionality and allows CA Deliver to capture output generated at a remote location. Network input is the supported way to capture SYSOUT generated on remote sysplexes.

If CA Deliver is run against a common database on multiple images within multiple sysplexes, the capturing, composing, and printing of reports and bundles now depends on where the bundle holding copy is kept.

- If the bundle holding copy is kept in JES, all the reports in a bundle must be generated and printed in that same sysplex. This means that the printer and remote definitions in all sysplexes have to be kept synchronized so that all output going to a specific printer or remote goes to the same destination regardless of the sysplex that prints the output.
- If the bundle holding copy is kept in CA View, you have to share the CA View databases across the sysplexes. The CA View databases also have to be shared across sysplexes if the reports are archived into CA View.

Before you attempt to share databases across sysplexes, verify that your GRS or equivalent environment can provide the proper protection for your databases.

Note: If hardware reserves are not preserved, all CA View and CA Deliver converted global enqueues must be propagated to all system images in all sysplexes.

How the Started Task Data Sets Are Allocated

The started task dynamically allocates SYSOUT data sets as they are needed for copies of reports.

It works this way:

- The report definition in the database is examined for print attributes that are used to allocate the data sets.
- Any attributes specified in the report definition are used "as is."
- For any attributes omitted from the report definition, the corresponding print attribute, as specified in the JCL for the DD statement, is obtained from the operating system SIOT and JFCB and used.
- If the initialization parameter OUTPUT is set to OUTPUT=YES, SJF services are invoked to obtain the print attributes in the first OUTPUT JCL statement, if any.
- During network input processing, the print attributes are obtained from JES.

Prior to producing any data for a report, the model banner page is read into memory and the starting banner page is constructed from the definition data for the report and output.

When the SYSOUT data set is closed and the task gains control:

- Records remaining in memory are written out for the reports
- The ending banner page is constructed and output
- The dynamically-allocated data sets are closed and (optionally) freed.

Remote Mirroring

The currently available DASD subsystems have the ability to create a real-time, remote mirror image of data. Both the CA Deliver database and checkpoint are candidates for mirroring.

If the CA Deliver checkpoint is mirrored, it is possible to see both RMOCPP05 and RMOCPP06 messages issued. The RMOCPP05 messages can be caused by long busy conditions caused by the hardware. These messages will stop when the long busy condition is dropped by the hardware.

Some additional information is listed below.

- The microcode that creates remote copies can elongate an I/O and produce what is called a 'long busy' condition.

We have seen 'long busy' conditions that last longer than 30 seconds. We have also seen DASD error recovery re-drive I/O after a 'long busy'.

- The 'long busy' conditions that are presented by the hardware can cause CA Deliver to issue RMOCPP05 messages.

When a channel program is started for the checkpoint, a timer is also started. If that timer expires without the I/O completing, the RMOCPP05 message is issued.

The RMOCPP05 messages stop when the 'long busy' ends.

- When a I/O is started, the signaling of its completion is normally accomplished with a single interrupt that gives both Channel End (CE) and Device End (DE) status.

We have observed that devices that are mirrored remotely many times are providing two I/O interrupts. The first is Channel End (CE) and the second is Device End (DE). This indicates that there can be longer I/O times as well as additional processor time that might be required to process the additional I/O.

Using a Test Database

We recommend that you use a test database and run CA Deliver as a separate task when you are following the procedures in this guide. This precaution protects you from inadvertently changing or losing data in your production database and prevents problems in your production environment.

Before setting up a unique, separate test system, you must do the following:

- Define a separate and unique database for your system.

- Assign a unique and separate identifier to your system.

Use the SYSID initialization parameter, which is a single letter, number, or national character (\$, #, @), to define the identifier of a system.

- Assign a unique and separate combination of network input attributes to your system if it is used for network input.

You must specify unique values for initialization parameters NETCLSL, NETDEST, and NETFORM).

- Assign a unique and separate combination of bundle holding copy attributes (BNDLCLS and BNDLDEST) to your system if it is to use the JES spool to retain bundle-holding copies.

- Create and maintain a unique and separate started task procedure and set of initialization parameters for your separate and unique system.

Using the SYSID Initialization Parameter

The SYSID initialization parameter establishes a unique identifier for each system.

The collating sequence of the identifiers determines the order in which the systems are offered control of SYSOUT. The default identifier is used if you do not specify a SYSID as \$ (\$ is the highest in the collating sequence).

Types of Report Processing

The following types of report processing are performed:

- Basic report processing
- Stacked report processing
- Interleaved report processing
- Monitored data output
- Control break report processing

Each type of report processing is described in this section.

Basic Report Processing

In basic report processing, there is a one-to-one correspondence between the SYSOUT data set and the report.

Only one report is defined for the SYSOUT DD statement. Records are written out as they are obtained.

Stacked Report Processing

Stacked reports are processed as follows:

1. Output records for a page accumulate in memory until either
 - The last line referenced in the report identification text has been passed to CA Deliver
 - The end of the page is reached.

A new page is defined by a skip to channel 1 carriage control.
2. Exclusive segments within the report identification text are examined in order by report for a match.
 - Any exclusive segment that has already been used and is not defined as reusable, is not examined, and is ignored.
 - The first exclusive segment for which a match occurs is marked as currently selected, and the exclusive segment previously marked as currently selected, if any, is unmarked (that is, not currently selected).
 - If matching text is not found, the exclusive segment previously marked as currently selected remains marked as currently selected.
3. The beginning definitions of overlapping segments within the report identification text are examined for matches.
 - Any overlapping segment that has already been used and is not defined as reusable is not examined and is ignored.
 - Any overlapping segment for which a match is found on its beginning definition is marked as currently selected.
 - Any overlapping segments already marked as currently selected remain marked as currently selected.
4. The page is written to any report with a currently selected segment.

If no reports have currently selected segments, the page is discarded (only if the undefined report identifier UNDEF, which is described later in this guide, is not used).

5. Currently selected segments are identified as used, and in the case of non-reusable segments, are no longer examined when searching for subsequent matches.
6. The ending definitions of overlapping segments within the report identification text are examined for matches.

Any overlapping segment for which a match is found is unmarked (that is, not currently selected).

Interleaved Report Processing

In interleaved report processing, multiple reports are associated with the same SYSOUT data set. The individual records of the reports are interleaved and contain a one-character interleave identifier that is used to separate the reports.

- If a SYSOUT data set does not contain carriage controls in its record format (for example, DCB=RECFM=FB), the interleave identifier is the first character of each record.
- If a SYSOUT data set contains carriage controls in its record format (for example, DCB=RECFM=FBA), the carriage control is the first character of each record and the interleave identifier is the second character of each record.

Example

Record 1 of report 1 is written, record 1 of report 2 is written, records 2 through 8 of report 1 are written, record 1 of report 3 is written, and so on.

- The interleave identifier of all of the records in report 1 contain the same character.
- The interleave identifier of all of the records in report 2 contain the same character (but different from report 1), and so on.

Specifications are defined in the report definition attributes to identify how the reports are separated. These specifications are referred to as *report identification text*.

As each record is processed, it is written to the appropriate report based on its interleave identifier. The interleave identifier is removed from the record before it is written. If no appropriate report is defined, the record is discarded.

Monitored Data Output

Monitored data output is a type of report processing in which only one report is defined for a SYSOUT data set.

In monitored data output, the SYSOUT data set is written out unchanged, exactly as defined in the JCL (or as specified when the DD statement is dynamically allocated).

Note: Banner pages are not produced for monitored data output.

Although the SYSOUT data set is not modified, the report is identified and tracked, and historical data is maintained. You can also use CA View to archive the data set automatically.

Monitored data output allows you to migrate to your first implementation of CA Deliver.

- You can use CA Deliver's utility to scan the job JCL
- You can subsequently construct and load the database with report definitions automatically.
- After the report definitions are loaded, you can track and archive reports immediately.

Control Break Report Processing

Individual pages within a data set are separated to produce unique reports.

Specifications are defined in the report definition attributes to identify the field that invokes the separator. These specifications are referred to as *control break identification*. When the contents of the defined control break field changes, a separator is printed between pages and processing continues until the contents change again.

Note: Reports of type CNTL honor the STKMODE initialization parameter. Therefore, if STKMODE=RECORD, the RECORD that contains the field defined is compared. If STKMODE=LINE, the LINE is selected for comparison in the print position as it appears on a page. If STKMODE is defined as LINE, carriage control is honored.

Obtaining Print Attributes

When the OUTPUT=YES initialization parameter is used, CA Deliver extracts print attributes from //OUTPUT statements and dynamically creates //OUTPUT statements as necessary to produce reports. The same rules used by the operating system for overrides are followed.

CA Deliver obtains print attributes for a report from the internal representation of the DD and OUTPUT JCL statements (from the JFCB, SIOT, and SWB control blocks).

Print Attributes

If a SYSOUT DD statement references more than one `//OUTPUT` statement, CA Deliver uses only the first statement to obtain print attributes. Other statements are ignored.

Print attributes for the following `//OUTPUT` parameters are obtained:

- ADDRESS
- BUILDING
- BURST
- CHARS
- CKPTLINE
- CKPTPAGE
- CKPTSEC
- CLASS
- COLORMAP
- COMPACT
- COMSETUP
- CONTROL
- COPIES=*nn*
- DATAK
- DEPT
- DEST
- DPAGELBL
- DUPLEX
- FCB
- FLASH
- FORMDEF
- FORMLEN
- FORMS
- FSSDATA
- INDEX
- INTRAY
- LINECT
- MODIFY

- NAME
- NOTIFY
- OFFSETXB
- OFFSETXF
- OFFSETYB
- OFFSETYF
- OUTBIN
- OUTDISP
- OVERLAYB
- OVERLAYF
- PAGEDEF
- PIMSG
- PORTNO
- PRMODE
- PRTERERROR
- PRTOPTNS
- PRTQUEUE
- PRTY
- RESFMT
- RETAINF
- RETAINS
- RETRYL
- RETRYT
- ROOM
- SYSAREA
- TITLE
- TRC
- UCS
- USERDATA
- USERLIB
- WRITER

FORMAT JES3 Control Statement Is Not Supported Directly

The `//*FORMAT` JES3 control statement is not directly supported by CA Deliver because the parameters on the statement are kept on the JES3 spool volume and are not available to CA Deliver.

Note:

- Although CA Deliver does not use the `//*FORMAT` control statement, JES3 can still apply the statement to a report DD statement that is dynamically allocated.
- Any `//*FORMAT` statement (that is, the `//*FORMAT` statement contains `DDNAME=`) is applied; however, only those parameters not specified by CA Deliver to allocate the DD statement dynamically are used.

It is unlikely, however, that any specific `//*FORMAT` statement (that is, the `//*FORMAT` statement contains `DDNAME=ddname`) is applied since the `DDNAME` of the dynamically allocated DD statement is generated by the operating system.

SMF Record Layouts

The Type 6 External Writer SMF record is optional and is created if the startup parameter `SMF=` is set to YES.

Note: The z/OS Type 6 record format definition can be found in the assembler language macro `SYS1.MACLIB(IFASMFR)`.

The Type 30 Common Address Space Work (generally used for accounting) is optional and can be created by the operating system and can be processed by CA Deliver to determine job termination.

- CA-Deliver uses these records if the startup parameter `SMF30=` is set to YES.
- If you specify NO, type 5 SMF records must be produced by your operating system for CA Deliver to determine job termination.

In a Cross Memory Services (XMS) session, SMF records can also be created. The creation and record type number is controlled by the XMS startup parameter `SMFSESS=`. The record format definition can be found in the assembler language macro `CAI.CVDEMAC(EBCSMFU1)`.

Note: SMF records can also be created by CA View. For more information about these SMF records, see the *CA View Reference Guide*.

Chapter 2: Initialization Parameters

You are provided with a set of parameters that initialize CA Deliver when you start the started task. This chapter describes these initialization parameters and presents them in alphabetical order.

This section contains the following topics:

[Overview of Initialization Parameters](#) (see page 35)

[BNDLMOUT, Attributes, and the First Report in a Bundle](#) (see page 37)

[Bundle Output Task and the BOT Initialization Parameter](#) (see page 37)

[Initialization Parameters](#) (see page 38)

Overview of Initialization Parameters

- You define initialization parameters in the data set named RMOPARMS.

You define the RMOPARMS data set in the start procedure JCL by specifying a DD statement named RMOPARMS. The RMOPARMS data set contains fixed, 80-byte records (blocked or unblocked).

Note: Sample member RMOPARMS in data set CAI.CVDEOPTN contains default initialization parameters.

- CA Deliver reads the information in positions 1 to 71 of the records in RMOPARMS. Information in positions 72 to 80 is ignored so you can use these positions for other purposes (for example, to contain sequence numbers).

Initialization Parameter Syntax

Syntax

parameter=value

where:

parameter

Defines the initialization parameter.

value

Defines the value to be assigned to the initialization parameter.

Syntax Rules

When specifying initialization parameters, do the following:

1. Specify one initialization parameter for each record.
2. Do *not* insert blank spaces between the initialization parameter and the equal sign or between the equal sign and the parameter value. The following is an example of a valid statement:

```
ARCH1=DIRECT
```

3. Continue an initialization parameter statement onto the next record by doing *one* of the following:
 - Move the statement so that its last character extends past column 71.
In this case, CA Deliver looks for the remainder of the statement in column 1 of the next record.
 - Insert a comma at the end of the record to be continued and enter the remainder of the statement in the next record.
Leading blanks in the next record are ignored.

Inserting Initialization Parameter Comments

To insert a record-long comment in RMOPARMS, enter an asterisk (*) in the first column. Characters after the asterisk are ignored.

To insert an inline comment, enter a space at the end of the initialization parameter statement.

Note: Characters after the space are ignored.

JOBCLSL Initialization Parameter

The JOBCLSL initialization parameter is *required*. You must specify a value for this parameter for CA Deliver to operate properly.

BNDLMOUT, Attributes, and the First Report in a Bundle

The initialization parameter BNDLMOUT works this way:

- If you set this parameter to NO or use the default for the started task, the attributes used to print a bundle (except for carriage control attributes) are obtained from the first report in the bundle that is written.

All the reports in the bundle are consequently printed with the same attributes.

- If you set this parameter to YES for batch bundling only, each report is printed with its own attributes except for destination and class.

Important: To accomplish this individual printing, a new DD statement is allocated and used whenever any attribute changes thereby generating several SYSOUT DD statements for the bundle.

To ensure that the SYSOUT DD statements print together, an OUTPUT JCL statement is dynamically created with the GROUPID parameter and is referenced by the DD statements.

Note: If a bundle distribution identifier is not defined in the bundle definition attributes, then the distribution identifier for the first distribution point in the bundle is used..

Bundle Output Task and the BOT Initialization Parameter

The BOT (Bundle Output Task) initialization parameter is a subtask of the started task that periodically scans the active bundle queue for bundles that are ready to print. The time period between scan actions is specified by the BNDLSCAN initialization parameter.

When CA Deliver runs on multiple operating systems in an installation, a bundle output task normally executes under CA Deliver on each operating system. BOT specifies whether the bundle output task is initially started. Also, operator commands are provided to start and stop the bundle output task.

Note:

- Running BOT on multiple operating systems that share the same database can affect performance.
- You can choose to not run the bundle output task on an operating system.

If a bundle is defined with JOB statements, the bundle output task submits a batch job to compose and print the bundle. The job that is submitted corresponds to skeleton JCL member RMOJCLB.

You can achieve parallel processing by submitting separate batch jobs to comprise the bundles. If no JOB statements are defined for a bundle, it is printed directly by the bundle output task.

Initialization Parameters

The following initialization parameters are described with their syntax and default values.

ARCH

The default archive criteria to be used by reports that are to be archived to CA View when no archive criteria value is specified in the ARCH field on the Report Definition Attributes panel.

Enter a value from 0–99 inclusive for nn (nn identifies and references archive criteria you specify with either ARCHnn initialization parameter described as follows).

For example, if you specify ARCH=2, archive criteria you specify for initialization parameter ARCH2 (ARCH2=//SAR/T, or ARCH2=DIRECT) are used by default for reports that are archived to CA View.

Note: Specifying 0 indicates that, by default, reports are not to be archived to the CA View database.

Syntax:

ARCH=*nn*

Default: 0

ARCHnn(Form 1)

ARCHnn(Form 1) specifies a set of criteria, that is identified and referenced by the value *nn*.

nn defines the attributes for reports that are to be archived to CA View using the JES spool.

You can specify a value from 1–99 inclusive for *nn* (or 01-99 depending on your installation option). This format specifies a set of SYSOUT attributes for using the JES spool as the intermediate holding area for reports to be sent to CA View for archiving. The set of SYSOUT attributes are:

- *c*— The SYSOUT class
- *ddddddd*—The destination
- *fffffff*—The forms name
- *m*—The archive medium, D (disk) or T (tape)
- *pppppppp*—The print mode

Note: The ARCH parameter can also be used with all attributes separated by slashes (/).

ARCHnn=c/ddddddd/fffffff/m/pppppppp

These attributes are positional in this format.

Note: If you do not specify an archive medium, reports are archived to disk by default.

For any report identified for archiving, the appropriate set of archival criteria is used to allocate a special SYSOUT data set to be spun off for archival by CA View.

Assume, for example, that your site has CA View installed and is archiving all SYSOUT with a forms name of SAR (the appropriate SAR initialization option is FORM=SAR). You then define two sets of archive criteria, numbered 1 and 2, for archiving reports to disk and tape as follows:

ARCH1=//SAR/D

ARCH1=(FORM=SAR,MEDIUM=D)

ARCH2=//SAR/T

ARCH2=(FORM=SAR,MEDIUM=T)

You can also define multiple specifications of an individual ARCHnn parameter for archival to multiple CA View databases.

Example

ARCH1=S///D

ARCH1=(CLASS=S,MEDIUM=D)

Archive using JES spool class S

ARCH1=T//SAR/D

ARCH1=(CLASS=T,FORM=SAR,MEDIUM=D)

Archive using JES spool class T with a forms name of SAR

Syntax:

ARCHnn=(CLASS=c,DEST=ddddddd,FORM=fffffff,MEDIUM=m,PRMODE=pppppppp)

Default: None

ARCHnn(Form 2)

Archive directly to a specific CA View database without passing the reports to the JES spool.

nn is a value from 1–99 (or 01-99 depending on your installation option).

You can specify **ARCHnn=DIRECT** to default to the database name specified in the **SAR=** initialization parameter or specifically reference the CA View database prefix.

For example:

ARCH2=DIRECT/PROD.VIEW

You can also define multiple specifications of an individual **ARCHnn** parameter for archival to multiple CA View databases.

Example**ARCH1=DIRECT**

Specifies archival to the database specified in the SAR= initialization parameter

ARCH1=DIRECT/TEST.VIEW

Specifies archival to the TEST.VIEW database

ARCH1=DIRECT/SAR1**ARCH1=DIRECT/SAR2**

Specifies archival to both the SAR1 and SAR2 databases.

Note: If the bundle holding copy is directed to this archival group, the holding copy resides in the first DIRECT database, that is, SAR1.

The DRAS Domain Name is optional and is only needed for the email notification feature.

The DRAS Domain Name identifies the ENF system identifier, DRAS server name, and View DRAS repository name that the OM Web Viewer is to use to access an email-attached report.

This domain name and the Web Server link (WEBSRVn initialization parameter setting) are components of the HTML WEBLINK that can be included within an email message. The format of the domain name is:

CCITASK.DRASTASK.DBNAME

- CCITASK— The ENF SYSID of the system where CCI is running
- DRASTASK— The name (unique identifier) of the DRAS task assigned on the DRASCFG configuration file SET SERVER ID(drasid) statement
- DBNAME— The repository name assigned to the View system by the DRAS task on the DRASCFG configuration file ACCESS REPOSITORY NAME(dbname) statement
- Example— *CCITASK.DRASTASK.DBNAME*

Syntax:

ARCHnn=DIRECT/database prefix

Default: None

AUTHTID

The suffix character to be used for the security table load module, RMOATHTx.

This module is used to authorize access to database data and functions.

If the started tasks for the same database are started on multiple systems, these started tasks must use the same AUTHTID setting.

Note: The RMODBASE utility always uses the default security table, RMOATHTB.

Syntax:

AUTHTID=x

Default: B

More information:

[Security](#) (see page 313)

AUTOACT

Specifies whether a report can be automatically activated by CA Deliver when opened.

If AUTOACT=NO is specified, CA Deliver excludes any report from processing that has not been previously activated.

Syntax:

AUTOACT=xxx

Default: YES

BANNER

The name of the default model banner page used to print non-bundled reports.

To specify that no banner pages are produced for reports as a default, do not specify a value for this initialization parameter (BANNER=).

Syntax:

BANNER=xxxxxxx

Default: BANNER

BEGINDAY

The time of day in a 24-hour format (1420 represents 2:20 p.m.) when a new day starts for bundle processing.

This parameter allows the bundles to remain open and active through the daily cycle time.

Syntax:

BEGINDAY=xxxx

Default: BEGINDAY

More information:

[TIME](#) (see page 78)

BNDLBNR1

The name of the default model banner page to be used for the bundle banner page when printing a bundle.

To specify that no bundle banner pages are produced for bundles as a default, code the BNDLBNR1 option with no value (BNDLBNR1=).

Syntax:

BNDLBNR1=xxxxxxx

Default: BDFLTB

BNDLBNR2

The name of the default model banner page to be used for the distribution banner page when printing a bundle.

To specify that no distribution banner pages are produced for bundles as a default, code the BNDLBNR2 option with no value (BNDLBNR2=).

Syntax:

BNDLBNR2=xxxxxxx

Default: BDFLTD

BNDLBNR3

Specifies the name of the default model banner page to be used to print the report banner page for a bundle.

To specify that no report banner pages are produced for bundles as a default, code the BNDLBNR3 option with no value (BNDLBNR3=).

Syntax:

BNDLBNR3=xxxxxxx

Default: BDFLTR

BNDLCLS

The SYSOUT class (cannot be a held class or a class assigned to a printer) to be used for the bundle holding copy of a report that is retained in the JES spool for bundling.

The SYSOUT class defined by BNDLCLS and the destination defined by BNDLDEST are used together to output a copy of the bundle to the JES spool.

The copy must be retained on the JES spool until the bundle is complete for printing. Therefore, the combination of BNDLCLS and BNDLDEST must be unique in the installation and must never be printed by the installation.

Note: When direct-to-View archival is in effect for a report to be bundled, no bundle holding copy of the report is created in the JES spool. Instead, the archival copy of the report on the CA View database is used when creating the bundle.

Syntax:

BNDLCLS=x

Default: 9

BNDLCONF

Specifies whether you want the bundle confirmation facility to be activated.

Note: Bundle confirmation can also be specified for an individual bundle on the Bundle Definition Attributes panel.

Syntax:

BNDLCONF=xxx

Default: NO

BNDLDEST

The SYSOUT destination to be used for the bundle holding copy of a report that is retained on the JES spool for bundling.

Important! The destination specified in this parameter must be defined to JES. Failure to do this can result in RMOPS102 or RMOPS202 messages and a U-0002 Abend in the bundle job.

Syntax:

BNDLDEST=xxxxxxx

Default: None

More information:

[BNDLCLS](#) (see page 45)

BNDLHDTL

Specifies whether accurate counts of lines and pages are to be available for bundled copies of reports.

This option requires that you set BNDLMOUT=YES and HDETAIL=YES in addition to installing the RMOJ2XIT, RMOFSSUX, RMOJ3X21 exits, as appropriate (depending on your environment).

Syntax:

BNDLHDTL=xxx

Default: NO

BNDLINT

The default bundling interval to be used for a bundle when no bundling interval is specified in the bundle definition.

Specify hours and minutes in 24-hour format (0220 represents two hours twenty minutes). If you omit this value or specify 0000, continuation bundles are not produced.

Note: This initialization parameter specifies a time interval, not the time of day.

Syntax:

BNDLINT=*nnnn*

Default: 0000

BNDLMOUT

Specifies whether a batch-submitted bundling job can use multiple DD statements to produce a bundle.

By using multiple DD statements, reports with different print attributes can be bundled but still printed with their different attributes. To ensure that the multiple DD statements are printed together, the bundling job uses the GROUPID parameter of the OUTPUT JCL statement.

Specify BNDLMOUT=YES if the following is true:

- The batch-submitted bundling job executes on a JES2 system
- You also specify OUTPUT=YES
- You specify &USERSET=YES in the JES2 initialization parameters

Syntax:

BNDLMOUT=*xxx*

Default: NO

BNDLSCAN

The time interval (in minutes) that the bundle output task waits between scans of the active checkpoint when it is looking for bundles to compose or submit.

Syntax:

BNDLSCAN=*nn*

Default: 2

BNDLWAIT

Specifies the default value for whether a bundle must wait for the end of a bundling interval before printing.

If BNDLWAIT=NO is specified (or defaulted), bundles do not wait but print as soon as all of the reports needed for the bundle are created (unless overridden in the bundle definition).

Syntax:

BNDLWAIT=*xxx*

Default: NO

BOT

Specifies whether the bundle output task is started at initialization time.

Syntax:

BOT=*xxx*

Default: YES

CCX

A set of user-defined carriage control specifications for carriage control identifiers *X*.

Specify any alphanumeric character (other than A or M) for *X*, the carriage control identifier.

- *c*—Specifies the carriage control character
- Specify *c* as either a one-character carriage control character or a two-character hexadecimal representation of the carriage control character.

Note: You must specify the characters blank, comma (,), and slash (/) in hexadecimal representation (40, 6B, and 61, respectively).

- *b1, b2, ..., b99*—Specifies the number of lines to skip (0 to 99) before writing the data
- *c1, c2, ..., c12*—Specifies the channel to skip to (1 to 12) before writing the data

To indicate that no data is to be written, specify an asterisk (*) where:

- *a1, a2, ..., a99*—Specifies the number of lines to skip (0 to 99) after the data is written or the total number of lines to skip when no data is written (when you specify an asterisk [*] for *b*)
- *c1, c2, ..., c12*—Specifies the channel to skip to (1 to 12) after the data is written or the channel to skip to when no data is written (when you specify an asterisk [*] for *b*)

For example, to define and assign the standard set of ASA carriage controls to the identifier U, specify the following:

CCU=40/1/0,0/2/0,-/3/0,+/0/0,

1/C1/0,2/C2/0,3/C3/0,4/C4/0,

5/C5/0,6/C6/0,7/C7/0,8/C8/0,

9/C9/0,A/C10/0,B/C11/0,C/C12/0

Syntax:

CCX=*c1/b1/a1,c2/b2/a2,...*

Default: None

DAYS

Specifies whether a new cycle automatically initiates on one of the seven days of the week (Monday through Sunday).

Y

Indicates that a new cycle is to be automatically initiated on a particular day.

N

Inhibits the automatic initiation of a new cycle on a particular day.

Syntax:

DAYS=xxxxxxx

Default: YYYYYYY

More information:

[TIME](#) (see page 78)

DEFDEL

Indicates whether recipients can delete the report from archive or alter the archive status within CA View.

Valid values are:

YES

Permits recipients to delete the report from the archive or alter the archive status.

NO

Prevents recipients from deleting the report or altering the archive status.

Note: The value set by DEFDEL either explicitly or implicitly is used to set the DEL value for new recipients added to Report Definition Distribution Specifications or Distribution List Specifications. This value does not change the settings of existing recipient definitions.

Syntax:

DEFDEL=YES / NO

Default: YES

DEFOUT

Indicates whether output is to be printed, or tracked or untracked for view only.

Valid values are:

YES

Creates printed output.

TRACK

Creates tracked, view-only output.

NO

Creates untracked, view-only output.

EMAIL

Creates EMAIL, view-only output

Note: The value set by DEFOUT either explicitly or implicitly is to be used to set the OUT value for new recipients that are added to Report Definition Distribution Specifications or Distribution List Specifications.

This value does not change the settings of existing recipient definitions.

Syntax:

DEFOUT=YES / TRACK / NO / EMAIL

Default: YES

DEFREPT

Indicates whether recipients can reprint reports within CA View.

Valid values are:

YES

Recipient can reprint the report.

NO

Recipient is not allowed to reprint the report.

Note: The value set by DEFREPT either explicitly or implicitly is to be used to set the REPT value for new recipients added to Report Definition Distribution Specifications or Distribution List Specifications. This value does not change the settings of existing recipient definitions.

Syntax:

DEFREPT=YES / NO

Default: YES

DEFRVIEW

Indicates whether the recipient has restricted viewing privileges for the report within CA View.

This parameter provides the ability to view through non-secured logical views and prevents the use of the VIEW command.

Valid values are:

YES

Recipient can only access the report with a non-secured logical view and is prohibited from using the VIEW command

NO

Recipient can access the report through any logical view and can use the VIEW command to create or change the logical view.

Note: The value set by DEFRVIEW either explicitly or implicitly is to be used to set the RVIEW value for new recipients that are added to Report Definition Distribution Specifications or Distribution List Specifications.

This value does not change settings of the existing recipient definitions.

Syntax:

DEFRVIEW=NO / YES

Default: NO

DYNRCHAR

The DYNRCHAR parameter identifies the replacement character that is used for invalid characters found in a dynamic report or distribution name.

The following characters are supported for a dynamic report or distribution name. The designated replacement character replaces all other characters.

- Letters(A-Z)
- Numbers(0-9)
- Period (.)
- National characters(\$, #, @)
- Percent (%)
- Cent (¢)
- Exclamation (!)
- Slash(/)
- Underscore(_)
- Dash(-)
- Blank ()
- Less than(<)
- Left parentheses“(“
- Plus (+)
- Bar(|)
- Ampersand (&)
- Right parentheses “)”
- Semicolon (;)
- Not Sign (¬)
- Broken bar (|)
- Comma (,)
- Greater than(>)
- Question mark (?)
- Colon(:)
- Single quote(')
- Equal (=)
- Double quote(“)

Syntax:

DYNRCHAR=*x*

Default: "."

EFORMAT

The name of the banner page that contains the default email format template. This template supplies the information that specifies where CA Spool is to deliver the message.

Note: This value can be overridden at the DISTID level and also on the report level.

Syntax:

EFORMAT= *xxxxxxxx*

Default: None

EMAILQ

The default Class, DEST, Form, and Writer for all email messages sent by CA Deliver are defined.

These values route the email message to a CA Spool email printer. To be sure that the values specified on the EMAILQ definition route the file to CA Spool for delivery, check the CA Spool parameter deck. There are several input interfaces used by CA Spool customers—one, or all of them are available at your site.

Note: You can use the EMAIL PARMS section of the email template to override these values.

Syntax:

EMAILQ=*class/dest/form/writer*

Default: None or "///"

EXTSEC

Specifies whether calls are to be made to external security.

- If you specify YES or UNIQUE, external security calls are to be made. The difference between them is in the resource names that are used.
- Specify EXTSEC=UNIQUE if you need separate rules for one or more regions. This causes external security calls to use the "RMO#" prefix, where the # sign is replaced with the region's SYSID= value.

Note: For EXTSEC=YES or EXTSEC=UNIQUE to be in effect, the started task must have been started at least once since the last IPL.

- If you specify NO, external security calls are not made.
- For EXTSEC=YES or EXTSEC=UNIQUE, you can also specify whether a list presented to a user contains all elements (NOLIMIT) or only those that the user is authorized to see (LIMIT).

The default is LIMIT.

Syntax:

EXTSEC=NO (YES,LIMIT) (YES,NOLIMIT) (UNIQUE, LIMIT) (UNIQUE, NOLIMIT)

Default: NO

EXTTRAN

If you use external security (RACF, CA ACF2 Security, CA Top Secret Security) and the report id contains characters from the extended special character set, specify this parameter as EXTTRAN=YES.

YES causes extended special characters to be translated automatically to ' _ ' underscores before the RACROUTE security call.

Syntax:

EXTTRAN=xxx

Default: NO

More information:

[Character Translation in Resource Names](#) (see page 328)

FEATURE

Occasionally, new features are added to CA Deliver and are given a number.

The features are activated with this parameter.

- To activate more than one new feature, enclose the numbers in parentheses and separate them with commas such as: `FEATURE=(1,7)`
- To deactivate all features, specify the feature parameter without any sub-parameter values, such as: `FEATURE=`

Syntax:

`FEATURE=`

Default: None

FEATURE=1

FEATURE 1 enables the Security RACROUTE WTO to produce informational messages which can be used to diagnose security problems.

- To enable, add number 1 to the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.
- To disable, remove the number 1 from the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.

Syntax:

`FEATURE=1`

FEATURE=2

Feature 2 enables the banner LRECL to be set to the report's LRECL.

- To enable, add the number 2 to the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.
- To disable, remove number 2 from the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.

Syntax:

`FEATURE=2`

FEATURE=3

Feature 3 disables checking for *cloned* Sysouts. If this feature is enabled, it is possible to archive multiple copies of a report if the JOBPARM COPIES field is set to a number greater than one.

- To enable, add the number 3 to the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.
- To disable, remove the number 3 from the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.

Syntax:

FEATURE=3

FEATURE=4

Feature 4 enables the special CBIS BUNDLE and UNBUNDLE code. This code processes bundles based on #BUNDLE or #UNBUNDLE placed in the special instruction fields.

- To enable add the number 4 to the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.
- To disable, remove the number 4 from the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.

Syntax:

FEATURE=4

FEATURE=5

Feature 5 enables the JES3 BNDLMOUT enhancement. This enhancement attempts to group bundled reports that contain compatible print attributes together.

- To enable add the number 5 to the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.
- To disable, remove the number 5 from the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.

Syntax:

FEATURE=5

FEATURE=6

Feature 6 enables an optional feature which is helpful for clients converting from CA Dispatch to CA Deliver.

When this option is enabled, CA Deliver does not count BLANK lines when processing a STACK report. Therefore if a STACK report is looking for something on line 1, it is interpreted as the first non-blank line.

- To bypass blank line counting, add the number 6 to the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.
- To disable, or count blank lines, remove the number 6 from the CA Deliver FEATURE startup parameter and recycle the CA Deliver started task.

Syntax:

FEATURE=6

FEATURE=7

FEATURE 7 allows the changes made by RMORRQUX to the fields ICBJNAM, ICBSNAM, ICBPNAM, and ICBJID to be saved.

These changed fields are used for the database search but are not copied to the Archival Print Attributes (APA) unless this feature is set.

Syntax:

FEATURE=7

FREEALL

Specifies whether all SYSOUT data sets created by CA Deliver are freed at close time and spun off for printing.

If you specify NO or you use the default, the only SYSOUT data sets freed at close time are those created by CA Deliver for a DD statement in which the FREE=CLOSE parameter is specified.

Syntax:

FREEALL=xxx

Default: NO

HDETAIL

Specifies whether the detailed historical data is maintained for reports and bundles.

- If you specify NO, only basic report data is provided.
- If you specify YES, history detail information are recorded for all reports and bundles (if BNDLHDTL is specified as YES).

To post lines and pages back to CA Deliver by JES, you must install the exits RMOJ2XIT, RMOFSSUX, and/or RMOJ3X21 as appropriate (depending on your environment).

Note: If you specify YES, history detail data can significantly increase the size of your database.

Syntax:

HDETAIL=xxx

Default: NO

HISTCNT

Specifies how the line and page counts in the basic history records are represented.

- HISTCNT=xxx (ALL/ONE) represents the line and page counts as follows:
- ALL – The total of all copies printed.
- ONE – The values for one copy of the report.

Syntax:

HISTCNT=xxx

Default: ALL

INBSSN

Specifies that CA Deliver is to interface with CA Balancing (CA Balancing Report Control).

Include the 4-digit or 4-character (alphanumeric) CA L-Serv Database Manager identifier (xxxx) that identifies the subsystem name of the CA L-Serv that manages the CA Balancing database.

Note: For more information about balancing, see your *CA Balancing* and *CA L-Serv* documentation.

Syntax:

INBSSN=xxxx

Default: None

JOBCLSL

Specifies a list of 1–38 classes of jobs to which report processing is limited.

This initialization parameter is required.

The job classes for TSO sessions and started tasks are specified as @ and \$, respectively.

Assume the following:

- That production jobs run in classes P and Q
- That production jobs are sometimes run using the same job name for test purposes in a class other than P and Q.

To limit processing only to the times the production jobs are run in production, specify the following statement:

JOBCLSL=PQ

Note: JOBCLSL works with the initialization parameter SYSCSL.

Syntax:

JOBCLSL=xxxxxxxx

Default: You must enter a value. An asterisk (*) means all classes are defined for the product.

JOBREF

Indicates whether pre-spool and post-spool collection must query the database for an alternate job definition based on the external writer or forms name of the SYSOUT data set.

An initial query is always performed based on the job name of the creating job. If that job name is not found in the database or if a matching step name, procedure step name, and data definition name is not found within the job definition, the alternate job definition can be retrieved and scanned for a matching step name, procedure step name, and data definition name.

Specify:

- WRITER to use the external writer name of the SYSOUT data set for referencing the alternate job definition
- FORM to use the forms name of the SYSOUT data set for referencing the alternate job definition.
- Blank to indicate that CA Deliver is not to query the database for an alternate job definition.

Syntax:

JOBREF=WRITER/FORM

Default: None

LOGO

The initial product logo panel is to display when you enter through an online interface.

Syntax:

LOGO=xxx

Default: YES

MAXHIST

The maximum number of generations of a report or a bundle of reports for which historical data is maintained.

Syntax:

MAXHIST=*nn*

Default: 3

MAXJESQ

The maximum time in minutes to wait for a bundle holding copy on the JES spool queued by JES and available for bundling.

The default value is sufficient in most cases.

Syntax:

MAXJESQ=*nnn*

Default: 10

NAME

The high-level, 1- to 17-byte name of the database.

Important! The database name is associated with the specified or defaulted SYSID. The database name must only be associated with one SYSID.

To prevent any residual effects, association changes must be made only when the system image is being IPLed.

Syntax:

NAME=*database prefix*

Default: None

NETCLSL

Specifies a list of 1–8 classes of SYSOUT data (*cccccccc*) to be used to select SYSOUT data sets for processing from the JES spool.

Follow the selection classes with a list of 1–8 new SYSOUT classes (*nnnnnnnn*), which positionally correspond to the selection classes assigned to the SYSOUT (only if a match for JOBNAME is not found in the database).

You must separate selection classes from new classes with a slash (/). The options NETCLSL, NETDEST, and NETFORM are used together to define a unique set of network input criteria. If you omit all three options, the product does not perform network input processing.

Note: NETCLSL must be unique within a single MVS image. Using network input attributes on multiple systems running the product is described in the *Administration Guide*.

Syntax:

NETCLSL=*cccccccc/nnnnnnnn*

Default: None

NETDEST

The selection destination (*ddddddd*) that is used to select reports for processing from the JES spool.

Follow the selection destination with the new destination (*nnnnnnnn*) assigned to the reports. You must separate the selection destination from the new destination with a slash (/).

Note: NETDEST must be unique within a single MVS image. Using network input attributes on multiple systems running the product is described in the *Administration Guide*.

Syntax:

NETDEST=*ddddddd/nnnnnnnn*

Default: None

NETFORM

The forms name (*ffffffff*) that is used to select reports for processing from other nodes in the network.

The selection forms name is followed by the new forms name (*nnnnnnnn*) assigned to the reports. The selection forms name is separated from the new forms name by a slash (/).

Note: NETFORM must be unique within a single MVS image. Using network input attributes on multiple systems running the product is described in the *Administration Guide*.

Syntax:

NETFORM=*ffffffff/nnnnnnnn*

Default: None

NETONLY

Specifies whether network input is the only method of operation for the product.

When you specify NETONLY=YES, the real-time operation of the product cannot be used since no dynamic interface is installed; when you specify NETONLY=NO, both the real-time and network input operations of the product can be used.

Syntax:

NETONLY=*xxx*

Default: NO

NETREQUE

Indicates which function the network input SAPI collector is to do:

- Requeue a JES SYSOUT data set
- Copy the JES SYSOUT data set to a new SYSOUT data set when the job name, step name, procedure step name, and ddname do not match database definitions

Note:

- When you specify NETREQUE=YES, the JES SYSOUT data sets are to be requeued to the new network input class, destination, and forms and retain the original job name, job number, and accounting information.
- When you specify NETREQUE=NO, JES SYSOUT data sets are to be copied to a new SYSOUT data set under the started task name.
- The NETREQUE initialization parameter is ignored if the NETUNDEF initialization parameter is set to YES and an UNDEF report is defined to the CA Deliver database. The NETREQUE initialization parameter is also ignored for cloned SYSOUT data sets that are created from jobs that have /*JOBPARM COPIES= specified.

Syntax:

NETREQUE=xxx

Default: YES

NETRERUN

Notifies the product that you are running CA 11 Workload Automation Restart and Tracking (CA 11) at a different destination.

- To indicate that you are not running CA 11 at a different destination, specify NETRERUN=NO
- To indicate that you are running CA 11 at a remote destination, specify NETRERUN=YES

Syntax:

NETRERUN=xxx

Default: NO

NETUNDEF

Indicates whether the network input collectors are to create UNDEF reports for JES SYSOUT data sets that do not match job name, step name, procedure step name, and ddname database definitions.

The UNDEF reports are only created if an UNDEF report is defined to the database.

- If you specify YES, UNDEF reports are created.
- If you specify NO, UNDEF reports are not created.

Syntax:

NETUNDEF=xxx

Default: YES

OFFPW

The password that a user must provide to withdraw CA Deliver from an operating system.

Syntax:

OFFPW=xxxxxxx

Default: No Password Required

OUTPUT

Specifies whether the product processes OUTPUT JCL statements.

You must specify OUTPUT=NO if your release of the operating system does not support the OUTPUT JCL statement.

Note: For more information about the OUTPUT JCL statement, see the chapter "Introduction."

Syntax:

OUTPUT=xxx

Default: NO

PASSTHRU

Specifies whether pre-spool processing must also pass the report through to the report's normal output specification.

When you specify PASSTHRU=YES, the report is dynamically intercepted and the report is allowed to be written to the normal output allocation. This feature provides a method for parallel testing. During the implementation of CA Deliver, the PASSTHRU=YES allows the reports to be processed normally and processed by CA Deliver. With PASSTHRU=YES, the output is written another time to the output DDNAME incurring additional CPU and I/O.

The default of PASSTHRU=NO causes the report to be dynamically intercepted and the original output DDNAME is not created.

Syntax:

PASSTHRU=xxx

Default: NO

PRBTASK

The number (*nn*) of external service requests that the product processes concurrently.

Enter a value between 1 and 16 inclusive for *nn*.

Syntax:

PRBTASK=*nn*

Default: 1

PREVRUN

Reports from the most recently run job for the product are to be flagged, deleted, or left unprocessed as follows:

The first value is for reports that have not been archived by CA View. The second value is for reports that have been archived. The second value takes precedence over the first value.

- FLAG is used in the exception code column to indicate that the report has been rerun.
- DELETE is used only when a report is stored in the CA View database or is a bundle holding copy in the JES spool.
- KEEP is used when a report is to be left unprocessed.

Explicitly specify the value for unarchived reports first, then a slash (/), then the value for archived reports, for example:

PREVRUN=FLAG/FLAG

Note: This initialization parameter works only if you set up the interface between CA 11 and CA Deliver. For more information about setting up this interface, see the *Installation Guide*.

Syntax:

PREVRUN=xxxxxx/xxxxxx

Default: KEEP/KEEP

REDISP

If set to YES, specifies that all users can display (refresh) online selection lists by pressing Enter instead of entering the REDISP input command on the command line (the default procedure).

Syntax:

REDISP=xxx

Default: NO or OFF

RMSWARN

Determines how RMORMS reacts to an inactive started task.

- When set to YES, RMORMS issues the RMORRS18 message, but it does not ABEND the job. Normal CA 11 restart processing continues.
- When set to NO, the default, RMORRS18 is issued, and the job does ABEND with code U0004.

This parameter is intended for use when the product is not active. Jobs not defined to CA Deliver need CA 11 processing, and RMORMS is executed to call CA 11 (U11RMS).

Note:

- With RMSWARN=YES, no CA Deliver restart processing is to occur for jobs defined to an inactive CA Deliver.
- To ensure that restart processing happens, set RMSWARN to NO.

Syntax:

RMSWARN=YES/NO

Default: NO

RPTENQ

Enables (YES) or disables (NO) exclusive ENQ to be used at the report level to serialize access to bundle holding copies residing on the JES spool.

Syntax:

RPTENQ=YES/NO

Default: NO

RPTNPROD

Specifies that a page containing the string *text-string* is to be output if no data is output for a stacked or interleaved report.

Note:

- You can use a maximum of 69 characters for *text-string*.
- You can include special and blank characters in *text-string*.
- If you include special or blank characters, you must enclose *text-string* in single quotation marks.

Syntax:

RPTNPROD=*'text-string'*

Default: None

SAR

The high-level name of the default CA View database for direct-to-View archival.

This database can contain the bundle holding copies if it is defined as the first DIRECT database in an ARCHnn parameter defined to the bundled reports.

Syntax:

SAR=*high-level-name*

Default: None

SARBUFCT

The number of buffers for archiving direct-to-View.

The number of buffers refers to the number of database blocks to be retained internally before writing on a report basis.

Important!

Memory is allocated from above-the-line storage in the application address space.

When processing dynamic reports that create thousands of reports, set SARBUFCT=1.

Syntax:

SARBUFCT=xxx

Default: 10

SECMSG

Specifying SECMSG lets you control notification of authorization failures.

- If you specify YES for the first parameter, the RMOATX01 message is sent to the SYSLOG.
- If you specify YES for the second parameter (valid only if external security is in use), the failure is logged with your external security manager.

Syntax:

SECMSG= (YES/NO, YES/NO)

Default: (NO,NO)

SETCMD

Enables (YES) or disables (NO) the SET operator command.

Note: The SETPW initialization parameter specifies the password that an operator must enter to use the SET operator command.

Syntax:

SETCMD=xxx

Default: NO

More information:

[SETPW](#) (see page 73)

SETPAGE

Specifies whether report output is to be overridden to a process mode (PRMODE) of PAGE.

- If you specify NO, the process mode (PRMODE) you specified is retained.
- If you specify YES, the process mode (PRMODE) is overridden to PAGE.

Syntax:

SETPAGE=xxx

Default: NO

SETPW

The password that an operator must enter to use the SET operator command.

You can specify up to eight characters.

Note: The SETCMD initialization parameter, which you use to enable and disable the SET operator command, is described earlier in this chapter.

Syntax:

SETPW=xxxxxxxx

Default: No Password Required

SMF

Specifies whether the product is to create special type 6 SMF records.

- If you specify NO, the product does not create special type 6 SMF records.
- If you specify YES, the product creates special type 6 SMF records.

Syntax:

SMF=xxx

Default: NO

SMF30

Specifies whether type 30 SMF records are produced by your operating system and are used by the product to determine job termination.

If you specify NO, type 5 SMF records must be produced by your operating system for the product to determine job termination.

Syntax:

SMF30=xxx

Default: NO

START

The name of the report activation procedure that is started at the start of a new daily cycle.

Syntax:

START=xxxxxxxx

Default: No Start Procedure

STKCHNn

The "skip to" line number for printer channels 2–12 for stacked report processing.

- Specifies the channel and has a value of 2–12 *line*
- Specifies the number of the line the printer skips to when receiving the skip-to-channel command

Syntax:

STKCHNn=*line*

Default: None

STKMODE

Indicates whether line or record mode processing is to be used to produce stacked reports.

LINE

Specifies that carriage control characters determine line spacing when searching for text to separate stacked reports.

RECORD

Specifies that carriage control characters are ignored (except for a skip to line one channel command) and each record is considered a line when searching for text to separate stacked reports.

In line mode, blank lines are counted; in record mode they are not.

Syntax:

STKMODE=xxxxxx

Default: LINE

STNAME_n

- Specifies the 1–8 character name of the 1–5 optional tracking stations for which detailed historical data for reports and bundles can be posted.
- Represents a value from 1–5 inclusive.

Syntax:

STNAME_n=xxxxxxx

Default: None

STKNOTXT

The default if no report identification text is specified.

ALL includes all report pages in the output

XCL includes report pages that are not explicitly matched by other exclusive reports

Note: Stacked reports without any text are treated as an exclusive report "catch-all" definition.

Syntax:

STKNOTXT=xxx

Default: XCL

STOPPW

The password that is required to stop the product on an operating system.

Syntax:

STOPPW=xxxxxxx

Default: No Password Required

SYSCLSL

A list of 1–36 classes of SYSOUT to which report processing is limited.

The product selects only those DD statements in the specified SYSOUT class.

For example, assume that your SYSOUT falls into classes S and R, and that production SYSOUT is sometimes produced using the same job name for test purposes in a class other than S and R.

To limit the product's processing only to the times the production jobs are run in production, specify:

`SYSCLSL=SR`

Additional information is presented in the description of the initialization parameter JOBCLSL earlier in this chapter.

Note: If you use CA Spool and CA Spool has been instructed to intercept the same class and destination that the product is intercepting, CA Spool intercepts the file. In this environment, be careful to select a SYSOUT class that is not being used by CA Spool.

Syntax:

`SYSCLSL=xxxxxxxx`

Default: All SYSOUT classes

SYSID

A unique identifier for the system.

The identifier consists of a single national character (\$, #, @, A–Z, 0–9).

The collating sequence of the identifiers determines the order in which the CA Deliver systems are offered control of SYSOUT.

The specified or defaulted SYSID is associated with database name specified by the NAME parameter. The database name specified by the NAME parameter must only be associated with one SYSID. To prevent any residual effects, association changes must only be made when the system image is being IPLed.

Syntax:

`SYSID=x`

Default: \$

TEXT

Specifies CAPS or ASIS according to whether report identification text is translated to uppercase (CAPS) when entered for a report definition or left as is (ASIS).

Syntax:

TEXT=xxxx

Default: CAPS

TIME

- The time of day in a 24-hour format (1420 represents 2:20 p.m.) when a new daily cycle is to be automatically initiated.
- Specify 0 to suppress the automatic initiation of any daily cycle.
- If you specify N for a particular day for the DAYS initialization parameter, the daily cycle on that day is not automatically initiated.

The daily cycle is triggered by an internal timer that is set based on the local time. When the local time changes because of daylight savings time, the internal timer is adjusted because the GMT offset has changed. Therefore the daily cycle starts at the proper local time after the time change has occurred.

Syntax:

TIME=nnnn

Default: 0

More information:

[DAYS](#) (see page 50)

If you specify YES, you are prompted to reply either Y or N to continue processing.

WARNING=xxx

Default: YES

WEBSVR_n

- The specific web server link that is to be used for each recipient is defined to the distribution definition.
- If none is specified, WEBSVR1 is used.

Example

WEBSVR1=HTTP:\\OMWEBSVR\\

Syntax:

[illegible]

Default: NO

WRITER

- **ASIS** – the **WRITER** field is populated from the **Writer** definition in **CA Deliver**.
- **DESTUSER** – The **WRITER** field is populated from the 'user' value of the 'dest.user' **Destination** definition in **CA Deliver**.

Syntax:

WRITER=ASIS/DESTUSER

Default: ASIS

Chapter 3: Operator Commands

This chapter explains how to issue the operator commands that are used to control CA Deliver and includes how to:

- Start, stop, and withdraw CA Deliver
- Refresh system routines
- Free reports and bundles from a system
- Start and stop the optional CA Balancing
- Start, post, and stop the bundle output task
- Reset the in-storage buffers
- Display currently active parameters on the console
- Set initialization parameters while CA Deliver is running
- Unlock the checkpoint
- Operate the Cross-Memory Task
- Define Multiple VTAM Cross-Memory Regions

This section contains the following topics:

[Starting](#) (see page 82)

[Refreshing System Routines](#) (see page 82)

[Stopping](#) (see page 83)

[Withdrawing](#) (see page 83)

[Freeing Reports and Bundles from a System](#) (see page 84)

[Starting and Stopping CA Balancing Report Control](#) (see page 84)

[Starting the Bundle Output Task](#) (see page 84)

[Stopping the Bundle Output Task](#) (see page 84)

[Posting the Bundle Output Task](#) (see page 85)

[Resetting the In-Storage Buffers](#) (see page 85)

[Displaying Currently Active Parameters](#) (see page 85)

[Setting an Initialization Parameter While the Product Is Running](#) (see page 86)

[Unlocking the Checkpoint](#) (see page 87)

[Operating the Cross-Memory Task](#) (see page 87)

[Defining Multiple VTAM Cross-Memory Regions](#) (see page 92)

Starting

To start CA Deliver on an operating system, issue:

```
S RMOSTC
```

where *RMOSTC* represents your STC name.

Note: After CAIRIM starts, add the command *S RMOSTC* to member *COMMNDxx* of *SYS1.PARMLIB* to start the task automatically at each IPL.

Refreshing System Routines

Certain CA Deliver system routines are not automatically reloaded when CA Deliver is started. System routines are service routines that are shared by all active CA Deliver started tasks on a given MVS system.

If SMP/E maintenance is applied to CA Deliver, new versions of the system routines can be reloaded by issuing the following start command:

```
S RMOSTC, PARM=REFRESH
```

where *RMOSTC* represents your STC name

Notes:

- Because the system routines are shared, only one CA Deliver started task needs to be restarted with the REFRESH parameter.
- If CA Deliver is started on multiple MVS systems, a refresh must be issued for each MVS system that is running CA Deliver.

When CA Deliver is started with the REFRESH parameter, an *RMOSVR03* or *RMOASR01* message will accompany each system routine that has changed.

Important! Do not use the REFRESH parameter as a standard startup parameter, especially when multiple release levels or maintenance levels are running on the same MVS system. This procedure could cause loading and use of back-leveled versions of CA Deliver's system routines.

Stopping

To stop CA Deliver on an operating system, issue:

```
P RMOSTC
```

where *RMOSTC* represents your STC name

Note: The execution of jobs monitored by the initialization parameters JOBCLSL, SYSCLSL, and/or NETCLSL, NETDEST, and NETFORM go into a wait mode until the task is started again.

Withdrawing

You can completely withdraw CA Deliver from an operating system at any time. When you withdraw CA Deliver, all dynamic interrupts from the system are removed, any service requests by jobs are canceled, and CA Deliver is terminated.

After you withdraw CA Deliver from a system, the distribution and tracking facilities continue to be performed on the current DD for each job in progress.

The consequences of withdrawing are:

- The DD statements currently being processed are not going to have ending banner pages
- History is not going to be completed for that report
- The checkpoint information might not reflect the true status.
- All subsequent DDs for the current step, and all subsequent steps are no longer under product control.
- CA Deliver distribution and tracking facilities are not performed on new jobs that you execute.

To withdraw CA Deliver that is active on an operating system, issue:

```
F RMOSTC,OFF
```

where *RMOSTC* represents your STC name

To withdraw the CA Deliver that is not active on an operating system (for example, after it is stopped), issue:

```
S RMOSTC,PARM=OFF
```

where *RMOSTC* represents your STC name

Freeing Reports and Bundles from a System

If an operating system fails, reports and bundles that were open or being processed on the failed operating system remain queued until you restart CA Deliver on that operating system.

To explicitly free queued reports and bundles so that you can submit them to other available operating systems, issue:

```
F RMOSTC, FREE xxxx
```

where RMOSTC represents your STC name, and xxxx specifies the system identifier of the operating system on which the reports and bundles to be freed are queued.

Important: *Do not* free reports for an operating system that is active.

Starting and Stopping CA Balancing Report Control

You must start and stop CA Balancing by issuing the INBSTC operator command in the CA Balancing started task.

Starting and stopping CA Balancing is described in the *CA Balancing System Guide*.

Starting the Bundle Output Task

To explicitly start the bundle output task on an operating system on which CA Deliver is running, issue:

```
F RMOSTC, BOT ON
```

where RMOSTC represents your STC name

Stopping the Bundle Output Task

To stop the bundle output task, issue:

```
F RMOSTC, BOT OFF
```

where RMOSTC represents your STC name

Posting the Bundle Output Task

At periodic intervals (defined by the BNDLSCAN initialization parameter), the bundle output task scans the active bundle queue for bundles that are ready to print.

To cause the bundle output task to wake up and scan the active bundle queue now, rather than waiting for the interval to expire, issue:

```
F RMOSTC,BOT POST
```

where RMOSTC represents your STC name

Resetting the In-Storage Buffers

CA Deliver maintains a number of buffers in storage. Issue the following command to refresh them:

```
F RMOSTC,RESET
```

where RMOSTC represents your STC name

This action causes a complete refresh of the following buffered data:

- Distribution Data Entries
- Banner Pages
- PRSET Entries
- Bundle Definition Table
- Jobname Translation Table

Important! If you change the Jobname Translation Table (referenced by the RMOJTAB DD statement), you must issue the F RMOSTC RESET command to refresh the in-storage copy of the table.

Note: CA Deliver itself issues the RESET command when any initialization parameter is changed with the SET operator command.

Displaying Currently Active Parameters

To display currently active parameters on the console (a series of write-to-operator messages with values in effect), issue:

```
F RMOSTC,DISPLAY
```

where RMOSTC represents your STC name

Setting an Initialization Parameter While the Product Is Running

To set an initialization parameter while CA Deliver is running, issue:

```
F RMOSTC,SET initparm=operand
```

where

RMOSTC represents your STC name

initparm represents the initialization parameter you want to set

operand represents the value to which you want to set the initialization parameter.

The initialization parameters that you can set with this operator command are listed below and are described in the chapter "Initialization Parameters."

ARCH	DAYS	SMF
AUTOACT	DYNRCHAR	START
BANNER	FREEALL	STKCHNn
BEGINDAY	GSS	STKMODE
BNDLBNRn	HDETAIL	STNAMEn
BNDLCONF	JOBCLSL	SYSCLSL
BNDLHDTL	LOGO	TEXT
BNDLINT	MAXHIST	TIME
BNDLWAIT	PASSTHRU	

Initialization Parameter Considerations

Be aware of the following:

- You *cannot* set any initialization parameters that are not included in the previous list. Initialization parameters and their operands are listed alphabetically in the chapter "Initialization Parameters."
- You must set the SETCMD initialization parameter to YES before this operator command can work.
- If you change the setting of the TIME initialization parameter, you will cause a recalculation of the daily cycle start time. This can impact the running of the daily cycle.

Unlocking the Checkpoint

You can unlock the checkpoint if the started task abnormally ends on a system, or if a system running CA Deliver fails while CA Deliver has the checkpoint locked.

Note: As long as the checkpoint is locked on a system, no other system can get it.

Never unlock a started task that shows the following message:

```
RMOSTC01 CA DELIVER IS INITIALIZED
```

Important! Under no circumstances should you unlock the checkpoint when the started task that holds the lock is still executing.

For example, if a started task hangs with the checkpoint locked, the started task could 'break free' and resume processing.

If the checkpoint is reset under these conditions and the started task resumes processing, the checkpoint could be damaged. If the checkpoint needs to be reset, cancel the started task that is hung and then issue the UNLOCK command on another CA Deliver started task executing on a different system image.

To unlock the checkpoint, issue the following command:

```
F RMOSTC,UNLOCK
```

where *RMOSTC* represents your STC name

Operating the Cross-Memory Task

The cross-memory task executes as the application for the cross-memory online retrieval option and is used for:

- The CICS pseudo-conversational option
- VTAM online retrieval
- IMS online retrieval
- ISPF/cross-memory
- TSO/cross-memory
- CA Roscoe/cross-memory

The standard procedure for the cross-memory task is located in CAI.CVDEOPTN.

Starting the Cross-Memory Task (START)

The cross-memory task is started with this operator command:

```
S RMOXMS
```

To assign a reusable ASID to the cross-memory task when it is started issue the following operator command:

```
S RMOXMS,REUSASID=YES
```

Note: The system honors a request for a reusable ASID only if REUSASID(YES) is specified in the parmlib member DIAGxx. Otherwise, the system assigns an ordinary ASID.

The PARM field in the RMOXMS PROC specifies an eight-character internal name for the system, the maximum number of users allowed, the transaction timeout limit, and a Cancel Connection field.

Immediate Termination (SHUTDOWN)

The cross-memory task is immediately terminated with the operator command:

```
F RMOXMS,SHUTDOWN
```

This command immediately terminates all active user sessions.

Normal Termination (QUIESCE)

A QUIESCE command causes termination of the cross-memory task when the last user ends his or her session. If no users are in session when the cross-memory online task is quiesced, the task terminates immediately.

The cross-memory task is quiesced with this command:

```
F RMOXMS,QUIESCE
```

or

```
P RMOXMS
```


Canceling Users (CANCEL)

Users are canceled from the cross-memory task by this command:

```

F RMOXMS,CANCEL [ALL]
                  [CONID=]
                  [UID#=]
                  [USERID=]
                  [SARDB=]
                  [RMODB=]
                  [INBDB=]

```

where:

CANCEL ALL

Cancels all users.

CANCEL CONID='connectid1','connectid2',... 'connectidN'<

Cancels a specific user using the 20-character connect-ID.

A user's connect-ID is displayed with the LIST command.

CANCEL UID#=nnn,nnn,nnn

Cancels a specific user using control block number.

A user's control block number is listed with the LIST command.

CANCEL USERID=userid1,userid2,userid3

Cancels a specific user.

CANCEL SARDB=database.name

Cancels all users accessing this CA View database.

CANCEL RMODB=database.name

Cancels all users accessing this CA Deliver database.

CANCEL INBDB=lservid

Cancels all users accessing this CA Balancing Report Control (CA Balancing) database.

Suspending Additional Logons (SUSPEND)

Additional user logons are suspended from cross-memory tasks using this command:

```
F RMOXMS,SUSPEND {ALL|SARVTAM|RMOVTAM|XMS|VTAM|SARDB=RMODB=|INBDB=}
```

Note:

- You can SUSPEND ALL, SARVTAM, RMOVTAM, XMS, or VTAM (both SARVTAM and RMOVTAM) tasks.
- If the VTAMPASS initialization parameter is set to YES, the VTAM interface attempts to pass the user to another task.
- You can also stop logons to a particular CA View, CA Deliver, or CA Balancing database with the SARDB=, RMODB=, and INBDB= parameters. If no users are logged on to the database, it is deallocated from the cross-memory region.

Resuming Acceptance of New Logons (RESUME)

The acceptance of new logons to the cross-memory task is resumed by this command:

```
F RMOXMS,RESUME {ALL|SARVTAM|RMOVTAM|INBVTAM|XMS|VTAM|SARDB=|RMODB=|INBDB=}
```

Notes:

- You can RESUME ALL, SARVTAM, RMOVTAM, INBVTAM, VTAM (all VTAMs), or XMS tasks.
- You can also resume acceptance of logons to a particular CA View, CA Deliver, or CA Balancing database with the SARDB=, RMODB=, and INBDB= parameters.

Closing Cross-Memory Tasks (CLOSE)

The CLOSE command closes the VTAM ACB and users are immediately terminated. Logons are not accepted until an OPEN command is issued, reopening the VTAM ACB.

CLOSE overrides the VTAMPASS parameter -- users attempting to sign on are not passed to another cross-memory task.

The command is:

```
F RMOXMS,CLOSE {ALL|SARVTAM|RMOVTAM|INBVTAM|VTAM}
```

You can CLOSE ALL, SARVTAM, RMOVTAM, INBVTAM, VTAM (all VTAMs), or XMS tasks.

Reopening Cross-Memory Tasks (OPEN)

The OPEN command reopens the ACB for the VTAM interface after a CLOSE command has been issued.

The command is:

```
F RMOXMS,OPEN {ALL|SARVTAM|RMOVTAM|INBVTAM|VTAM}
```

You can OPEN ALL, SARVTAM, RMOVTAM, INBVTAM, VTAM (all VTAMs), or XMS tasks.

Listing Statistics (LIST)

A listing of information relating to the cross-memory sessions is obtained by issuing this command:

```
F RMOXMS,LIST [DATABASE]
               [STATUS]
               [USERS]
               [USERID=]
               [UID#=]
               [CONID=]
               [SARDB=]
               [RMODB=]
               [INBDB=]
```

where:

LIST DATABASE{,ALL}

Displays the cross-memory system status and active databases for this task;

Use the ALL option for all tasks.

LIST STATUS{,ALL}

Displays the cross-memory system status of this task.

Use the ALL option for all tasks.

LIST USERS{,ALL|,ACTIVE|,INACTIVE}

Displays the status of ALL, ACTIVE, or INACTIVE users.

Default: ACTIVE.

LIST USERID='userid1',

'userid2','userid3'

Displays the status of a specific user.

LIST UID#=nnn,nnn,nnn

Displays the status of a specific user who is identified by control block number.

LIST CONID='connectid1','connectid2','connectid3'

Displays the status of a specific user who is identified by the 20-character connect-ID.

LIST SARDB=databasename

Displays all user sessions using the specified CA View database.

LIST RMODB=databasename

Displays all user sessions using the specified CA Deliver database.

LIST INBDB=lservid

Displays all user sessions using the specified CA Balancing database.

Changing Cross-Memory Parameters Online (CHANGE)

You can use the CHANGE command to modify these cross-memory initialization parameters online:

CANCEL
LONGWAIT
USERMAX
VTAMPASS

Important! You cannot increase USERMAX above the value it had when RMOXMS was started, or set VTAMPASS from NO to YES; the region must be recycled.

To change the USERMAX parameter, enter:

F RMOXMS,CHANGE USERMAX=nnn

Defining Multiple VTAM Cross-Memory Regions

The *Installation Guide* contains the steps to follow to install the cross-memory task. (VTAM online retrieval runs under the cross-memory task.) The following sections provide guidelines for defining multiple VTAM/cross-memory regions.

Regions Grouped by Operating System Subsystem Name

Multiple VTAM regions are grouped under a four-character operating system subsystem name. The SUBSYS= cross-memory parameter specifies the subsystem name at the startup time of the cross-memory region.

Unless you specifically want to define separate multiple groups of VTAM regions, we suggest that you use the default SUBSYS parameter.

Note: If you specify SUBSYS=, all VTAM regions grouped under the same SUBSYS *must* be started with the same release/PTF level of the CA View, CA Deliver, and/or CA Balancing load libraries.

Important! If you change your release of CA View, CA Deliver, and/or CA Balancing and you specified a value in SUBSYS=, you must change the SUBSYS= cross-memory parameter and IPL to prevent CSA damage.

The control block sizes for CA View, CA Deliver, and/or CA Balancing are release-dependent and serious problems can result if you do not specify the correct release.

If you use the default parameter for SUBSYS, you will not have this problem because each release of CA View, CA Deliver, and/or CA Balancing defaults to the appropriate SUBSYS name.

Regions Separated by REGIONIDs and APPLID

Be aware of the following:

Each separate VTAM region attached to a SUBSYS must have a unique REGIONID.

- You must specify the REGIONID as the first parameter of the PARM= statement in the start procedure JCL for the cross-memory online retrieval task.
- A region does not start if another region with the same REGIONID is already running for the same SUBSYS.

Each separate VTAM region attached to a SUBSYS must have a unique VTAM APPLID.

The USSTAB (defined in Step 2 of Installing the VTAM Online Retrieval Option in the Installation Guide) or your session manager should pass all VTAM logon requests to the Primary VTAM APPLID.

Notes:

- CA View reroutes requests to other regions if the Primary is full or marked as unavailable.

Each VTAM APPLID must have AUTH=(ACQ,PASS) and you must specify the cross-memory parameter VTAMPASS=YES for all of the regions.

Checking the Status of Multiple Regions

Enter the LIST STATUS operator command with the ALL parameter to display the status of all regions in the same SUBSYS group. The command is:

```
F RMOXMS,LIST STATUS,ALL
```

Specifying the Parameters: SYSIN Statements

You should specify as many of the parameters as possible with SYSIN statements to manage multiple regions easily.

The REGION parameter is positional and must be specified in the PARM= statement of the cross-memory task JCL.

You should specify the following parameters in the PARM= statement:

SUBSYS=
RMOAPPL=
XMS=

You can specify the rest of the parameters with SYSIN DD statements. If you place the SYSIN statements in a PDS member, you can alter the parameters without shutting down the cross-memory region; however, the parameters do not take effect until the next time the region is shut down and restarted.

Chapter 4: Utilities

This section contains the following topics:

[What Is a Control Statement?](#) (see page 97)
[Saving Time with Conversion Utilities RMODBB and RMOJCL](#) (see page 98)
[RMOBPR - Batch Bundle Posting](#) (see page 99)
[RMOCPMAP - Checkpoint Map Utility](#) (see page 100)
[RMODBASE - Database Maintenance](#) (see page 104)
[RMODBB - Database Construction from Existing Data](#) (see page 127)
[RMODSC Report Data Set Collector](#) (see page 163)
[RMOGRW - General Report Writer](#) (see page 167)
[RMOHTP - Batch Detail History Reporting](#) (see page 225)
[RMOIFMAP - Index File Mapping Utility](#) (see page 229)
[RMOJCL - Automatic Database Construction From JCL](#) (see page 229)
[RMOJCS - Enhanced Database Construction from JCL](#) (see page 234)
[RMOPRE - Take Action on the Most Recently Produced Reports](#) (see page 239)
[Using Page and Form Definitions](#) (see page 243)
[RMOPSF - Form and Page Definition Modification](#) (see page 246)
[RMORAP - Activating and Deactivating Reports from Batch](#) (see page 250)
[RMORMS - Using Tracking](#) (see page 255)
[RMORPT - Batch Reporting](#) (see page 258)
[RMORXB - Rebuilding Cross-Reference Records](#) (see page 263)
[RMOUTIL - Migration Support](#) (see page 264)

What Is a Control Statement?

A *control statement* is a parameter that defines, augments, or modifies how a job control statement operates. You specify control statements in a card image data set.

Example

The BLOAD control statement, which you use with the RMODBASE utility program, loads the model banner page members in the model banner page library to a database.

Syntax

The syntax of a control statement can take one of two forms:

`control-statement operand operand ...`

`/control-statement operand operand ...`

Syntax Rules

The rules for using control statements are as follows:

- Place a control statement that does not require a slash in a single card image anywhere between columns 1 and 71 (inclusive).
Note: You can precede a control statement that does not require a slash with one or more blanks.
- For control statements that require a slash:
 - Place the slash in column 1.
 - Place the control statement anywhere between columns 2 and 71 (inclusive).
 - Do not use a slash on continued lines.
- Insert one or more blanks between a control statement and the first operand.
- Insert one or more blanks or commas between each subsequent operand.
- Use a new card image to specify control statements that span more than one line. A control statement logically continues from column 71 of the preceding line to column 1 of the next line.
- Place an asterisk in column 1 of a card image to specify a comment. All text on the line following the asterisk is treated as a comment.

Saving Time with Conversion Utilities RMODBB and RMOJCL

Perform the following list of tasks when setting up your database for report processing:

1. Define the distribution identifiers.
2. Define the jobs.
3. Define the text specifications for separating reports.
4. Add distribution identifiers to the reports.
5. Modify report attributes, if necessary.

Note: You can eliminate many of the preceding tasks by using the conversion utilities RMODBB and RMOJCL in CAI.CVDEJCL.

How to Quickly Set Up a Database

To set up your database quickly, follow these steps:

1. Run the RMOJCL utility to construct the job and report definitions in the database.
2. Run the RMODBB utility to define the distribution identifiers and add them to the reports.

3. Review and modify the definitions in the database, and specify text specifications for separating report data, special instructions, and other special print attributes.
4. When you are ready to place a report under full control of CA Deliver, change the type of report processing in the job definition from monitored data output to the type of report processing you want.

For more information about conversion utilities, see the *Administration Guide*.

RMOBPR - Batch Bundle Posting

Use the RMOBPR batch utility program to post one or more bundles for printing.

Important! The started task must be executing on the same operating system as all batch and online facilities that access checkpoint data detail, historical data, and facilities you use to delete definitions for these batch and online facilities to work.

Example

To post bundles CASH1, ACCT275, and ACCT277 for printing, execute the following job, which is located in RMOBPR in CAI.CVDEJCL:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//BUN EXEC PGM=RMOBPR,PARM='DELIVER.SYSTEM1,CASH1,ACCT275,ACCT277'
//RMOBID DD DSN=DELIVER.BIDLIB,DISP=OLD
```

Job Control Statements

Specify the following JCL to execute RMOBPR:

JOB

Initiates the job.

EXEC

Specifies the name of the program (PGM=RMOBPR) and the parameters (PARM=) for the program.

Use the PARM parameter to specify the high-level name of a database and at least one or more bundle identifiers which you use to identify the bundles you want to print. For example:

PARM='DELIVER.SYSTEM1,BID1,BID2'

Separate each bundle identifier with a comma.

The PARM parameter can be 100 characters in length.

RMOBID DD

Defines the sequential data set that contains the bundles you want to print.

Specify one bundle identifier per record. You can specify bundle identifiers either with this job control statement or through the EXEC job control statement parameters.

This job control statement is optional.

RMOCPPMAP - Checkpoint Map Utility

CA Technical Support uses the RMOCPPMAP diagnostic utility to determine how the space in the checkpoint is being used. The output contains a breakdown of the type of records in use and amount of free space.

Sample JCL

To run RMOCPMAP, specify the following JCL:

```
//... JOB ...
//STEP1 EXEC PGM=RMOCPMAP
//STEPLIB DD DISP=SHR,DSN=CAI.CVDELOAD <=== Deliver load library
//RMOCPIN DD DISP=SHR,DSN=COPY.OF.DELIVER.CHECKPT
//SYSPRINT DD SYSOUT=*
```

The input specified on the RMOCPIN DD statement must be a copy of the checkpoint. You can create this input with IEBGENER as follows:

```
//... JOB
//STEP1 EXEC PGM=IEBGENER
//SYSUT1DD DSN=dlvrdb.hlq.RMODBASE.C0000001,DISP=SHR
//SYSUT2DD DSN=copy.of.deliver.checkpt,disp=(,catlg),
// UNIT=uuuu,VOL=SER=vvvvvv,SPACE=(CYL,(cc)),
// DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096)
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
```

Note: The space allocation must match that of the checkpoint file.

Checkpoint Space Analysis

RMOCPMAP produces the following report on SYSPRINT:

CA DELIVER CHECKPOINT SPACE ANALYSIS

	RECORDS	BYTES			
FREE BLOCKS	882	3,612,672			49%
FREE CRJ	24,861	1,292,772			17%
FREE CRB	15,949	318,980			4%
RCE	898	21,552	-	-	-
CRJ	26,697	1,388,244			18%
CRB	31,991	639,820			8%
BCE	33	1,188	-	-	-
UNUSED RCE	1,992	47,808	-	-	-
UNUSED BCE	80	47,808	-	-	-
CLR BLOCKS	4	16,384	-	-	-
OVERHEAD		30,500			
TOTAL		7,372,800			
TOTAL IN FILE		7,372,800			
DIFFERENCE		0			
STATUS (AS REPORTED BY RMODBASE)					
CYLINDERS:		20			
BLOCKS:		1,800			
USED BLOCKS:		918			
PERCENT USED:		51%			

Types of Records in a Checkpoint

To interpret the Checkpoint Analysis Report, you must know the kinds of records that are kept in the checkpoint.

RCE Checkpoint Report Entry

Describes reports on the Active Report List panels and points to the following options:

CRJ

Checkpoint Report Job Entry.

There is one CRJ per job that contains the report.

CRB

Checkpoint Report Bundle Entry.

There is one CRB per report/job contained in the bundle.

An RCE is used when a new report ID is activated.

CRJs and CRBs are used when a new instance of a report is activated.

There is one RCE Checkpoint Report Entry per active report ID.

BCE Checkpoint Bundle Entry

Describes bundles on the Active Bundle List panel.

A BCE is used when a new bundle is activated.

There is one BCE Checkpoint Bundle Entry per active bundle.

Number of Records in a Checkpoint

A checkpoint block can hold the following number of records:

Record	Number
RCE	92
BCE	70
CRJ	54
CRB	97

Checkpoint Contents Report

A report of the contents of the checkpoint is also available. Specify PARM=D on the EXEC statement.

Note: The Checkpoint Contents Report is a dump of the checkpoint and can be large and is intended for use by CA Technical Support.

RMODBASE - Database Maintenance

Use the RMODBASE utility program (located in CAI.CVDEJCL) to define and maintain the database. Specifically, use RMODBASE (which runs authorized) to do the following actions:

- Define a new database
- Add additional space to a database
- Create and copy the checkpoint data set
- Copy a database
- Rename a database
- Delete a database
- Provide usage statistics on a database
- Load the online library to a database
- Load the model banner page library to a database
- Load the printer setup library to a database
- Unload a database
- Load a previously unloaded database
- Delete history data from a database
- Convert older versions of records to a record format compatible with the latest version of CA Deliver

Important! RMODBASE always uses the default security table, RMOATHTB.

Job Control Statements

Specify the following JCL to execute RMOBASE:

JOB

Initiates the job.

EXEC

Specifies the name of the program (PGM=RMOBASE) and optional parameters (PARM=) for the program.

Use the optional PARM parameter to specify the high-level name of a database (PARM='DELIVER.SYSTEM1').

STEPLIB DD

Identifies the load library that contains RMOBASE.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

RMOOLIB DD

Identifies the online library that contains the panel, message, and skeleton JCL members you want to load into the database.

Specify this job control statement only if you use the OLOAD Control Statement, which is described later in this chapter.

RMOBLIB DD

Identifies the model banner page library that contains the model banner pages you want to load into the database.

Specify this job control statement only if you use the BLOAD Control Statement, which is described later in this chapter.

RMOPLIB DD

Identifies the printer setup library that contains the printer setup members you want to load into the database.

Specify this job control statement only if you use the PLOAD Control Statement, which is described later in this section.

RMOLOAD DD

Identifies a sequential input data set that contains the unloaded database you want to load.

Specify this job control statement only if you use the LOAD Control Statement, which is described later in this chapter.

RMOCONV DD

Identifies the sequential input that contains the unloaded copy of the Release 11 database that is being converted to Release 11.5.

RMOUNLD DD

Identifies a sequential output data set to which you want to unload the database.

Specify this job control statement only if you use the UNLOAD Control Statement, which is described later in this chapter.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

Be aware of the following conditions:

- You can execute CONVERT, COPY, DELETE, HDELETE, LOAD, MAKECKPT, RENAME, and VERSION against the database specified by the NAME control statement only when the started task is not executing.
- You can execute ADDDS, BLOAD, OLOAD, PLOAD, UNLOAD, and STATUS against any database, whether the started task is executing or not.

Note: ADDDS, BLOAD, and OLOAD are required for a new install.

ADDDS Control Statement

The ADDDS control statement is used to create a space or add additional space to an existing database. Space is added by creating a data set and formatting it with fixed-length blocks. First, define the high-level name of the database with the NAME control statement (or the PARM parameter of the EXEC JCL statement). ADDDS optionally supports IBM's System Managed Storage (SMS).

Syntax for a Non-SMS Data Set

```
ADDDS UNIT=unit VOLSER=volser CYLINDER=cyls BLKSIZE=b ABOVE
```

Syntax for an SMS Data Set

ADDDS CYLINDER=*cyls* BLKSIZE=*b* DATACLAS=*dc* MGMTCLAS=*mc* STORCLAS=*sc* ABOVE

where:

ABOVE

Indicates that the data set being added to the database can be allocated above the 64K cylinder line on a 3390-A device. Data sets that are allocated above the 64K cylinder line are allocated in increments of 21 cylinders, so the final data set allocation is rounded up to the next multiple of 21 cylinders.

This parameter is optional.

unit

Specifies the unit name to be used to dynamically allocate a new data set for the database.

This operand is optional.

volser

Specifies the volume serial number on which to allocate the new data set for the database.

This operand is optional.

cyls

Specifies the number of contiguous cylinders to allocate to the data set for the database, between 1 and 4369.

This operand is required.

b

Specifies the block size of the data set for the database, between 3768 and 16383 bytes.

This operand is optional and if you leave this operand blank, the default (3768 bytes) is used. This operand is valid only for the first extent of the database.

Note: When converting databases, if you specify a value for this operand that is smaller than was originally specified for the "from" database, specify a value for the delete history operand in the COPY or LOAD control statement.

dc

Specifies the SMS data class you want to assign to the data set for the database.

This operand is optional and works only if you have IBM's SMS.

mc

Specifies the SMS management class you want to assign to the data set for the database.

This operand is optional and works only if you have IBM's SMS.

sc

Specifies the SMS storage class you want to assign to the data set for the database.

This operand is optional and works only if you have IBM's SMS.

Interdependency of ADDDS Parameters

The following table summarizes the interdependency of the ADDDS parameters for SMS and non-SMS data sets; it illustrates the parameters that you must use, parameters you can optionally use, and parameters you do not specify.

Parameter	Data Set Not Managed by SMS	Data Set Managed by SMS
BLKSIZE	Optional	Optional
CYLINDER	Required	Required
DATACLAS	Do not specify	Optional
MGMTCLAS	Do not specify	Optional
STORCLAS	Do not specify	Optional
UNIT	Optional	Optional
VOLSER	Optional	Optional

Calculating Cylinders for Your Database

There is no exact method for computing the number of cylinders required for a database. You can, however, use the formula in this section initially to calculate the number of cylinders to set aside for the database.

Note: The amount of space you actually require can vary significantly from the value you calculate. Use the STATUS control statement periodically to determine the actual space you use and adjust the number of cylinders accordingly.

Use the following formula to calculate cylinders for your database:

1. Add the following numbers:
 - The total number of jobs that produce reports
 - The total number of different reports that can be produced
 - The total number of individuals to which reports are distributed (the number of distribution identifiers)

2. Divide the sum of the preceding numbers by 250 and then round the result up to the next integer.
3. If you plan to maintain detailed historical data (set the initialization parameter HDETAIL to YES), do the following:
 - Add the total number of different reports that can be produced to the total number of different bundles that can be produced.
 - Multiply the result by the maximum number of generations of historical data you plan to maintain (the value you specify for the initialization parameter MAXHIST).
 - Divide the product by 195 and round the value up to the next integer (195 is the blocks per cylinder on an IBM 3390 at a block size of 3768).
4. The value you calculate is the number of cylinders to use (do not use fewer than three cylinders).

BLOAD Control Statement

The BLOAD control statement is used to load the model banner page members in the model banner page library to the database. The model banner page library is defined with DD statement RMOBLIB. The model banners were installed into CAI.CVDED133 during the SMP Apply step.

The high-level name of the database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

BLOAD

Note: A sample utility program, UNLBANEX, included in CVDEOPTN, can extract the CA Deliver banners from an unloaded copy of the CA Deliver database and create members of a partitioned data set with the same member names as the banner names in the CA Deliver database. For more information, see the topic UNLBANEX - Sample Banner Extraction Utility.

UNLBANEX - Sample Banner Extraction Utility

UNLBANEX is a sample program to extract banners from an unloaded copy of a CA Deliver database and create PDS members.

Assemble and link UNLBANEX to CVDELOAD:

```
//      JOB
//ASM    EXEC PGM=ASMA90,PARM=( 'NODECK,OBJECT' )
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//SYSUT2 DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//SYSLIN DD DSN=&&OBJ,DISP=(,PASS),
//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//SYSLIB DD DISP=SHR,DSN=SYS1.MACLIB
//SYSIN  DD DISP=SHR,...CVDEOPTN(UNLBANEX)    <== Change
//LKED   EXEC PGM=IEWL,PARM='XREF,LIST,NCAL'
//SYSLMOD DD DISP=SHR,DSN=...CVDELOAD        <== Change
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSLIN DD DSN=&&OBJ,DISP=(OLD,DELETE)
//      DD *
//          ENTRY  UNLBANEX
//          NAME    UNLBANEX(R)
/*
//
```

Unload the CA Deliver database using the RMOBASE/UNLOAD utility and reformat the RMODBASE VBS output file to normal Fixed Blocked Variable records.

Note: When unloading the Deliver database, specify the HISTDELALL operand, as unloading the history detail creates an unnecessary large file.

```
//UNLOAD EXEC PGM=RMOBASE
//STEPLIB DD DISP=SHR,DSN=...CVDELOAD        <== Change
//SYSPRINT DD SYSOUT=*
//RMOUNLD DD DSN=...UNLOAD,                  <== Change
//          DISP=(,CATLG),
//          SPACE=(TRK,(100,100)),UNIT=SYSDA
//SYSIN   DD *
//          NAME ....db_hlq.....              <== Change
//          UNLOAD HISTDELALL
/*
//FORMAT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=...UNLOAD          <== Change
//SYSUT2 DD DSN=...UNLOADR,                  <== Change
//          DISP=(,CATLG),
//          SPACE=(TRK,(100,100)),UNIT=SYSDA,
//          DCB=(,RECFM=VB,LRECL=32756,BLKSIZE=32760)
```

```
//SYSIN DD DUMMY
//
```

Pre-allocate a partitioned data set with the same file attributes as CVDED133 (CA Deliver Banner Library):

```
DSORG=PO
RECFM=FB
LRECL=133
BLKSIZE=3990
```

Run the UNLBANEX utility to create a member for each banner in the original CA Deliver database:

```
//RUN EXEC PGM=UNLBANEX
//STEPLIB DD DISP=SHR,DSN=...CVDELOAD <== Change
//SYSPRINT DD SYSOUT=*
//RMOBLIB DD DISP=SHR,DSN=...PDS <== Change
//RMOUNLD DD DISP=SHR,DSN=...UNLOADR <== Change
//
```

CONVERT Control Statement

The CONVERT control statement is used to convert an older release level of the database to the current release level. The CONVERT control statement can be used to convert a Release 1.6 or 1.7 database to Release 11.5.

A sequential data set containing an unloaded copy of the old database is provided as input through the RMOCONV DD statement. The unloaded copy of the database is created with the older release level version of the RMODBASE utility UNLOAD control statement. The old database is converted to the current release level format and loaded to the database defined previously on the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Note: After converting the database, the current release level online panels has to be loaded into the database with the OLOAD control statement of the RMODBASE utility.

Syntax

CONVERT *buffer-option reserve-option history-option*

where:

buffer-option

Specifies either BUFFER, to buffer records written to the output database that improves performance and reduces input/output time, or NOBUFFER to eliminate buffering.

If omitted, NOBUFFER is used by default.

Note: When using BUFFER, an unsuccessful completion could corrupt the output database.

reserve-option

Specifies either RESERVE, to place a reserve on the output database for the duration of the convert operation, or NORESERVE, to not place a reserve on the output database

Specify NORESERVE only if you are certain that no one else could be updating the output database while the CONVERT operation is in process.

If omitted, RESERVE is used by default.

history-option

Deletes detail history when you convert a database

Specify either HISTDEL to delete the detail history subfile only, or HISTDELALL to delete the history subfile and all basic history records.

COPY Control Statement

The COPY control statement is used to copy a database. The output database must be a newly created, empty database.

The COPY control statement can also be used to reorganize a database and reclaim blocks in the database that are allocated but not currently being utilized.

Note: To merge databases, use the RMOUTIL utility.

The high-level name of the output database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

COPY from-name reserve-option buffer-option history-option

where:

from-name

Specifies the high-level name of the database to be copied.

This operand is required.

reserve-option

Specifies either RESERVE, to place a reserve on the output database for the duration of the copy operation, or NORESERVE, to not place a reserve on the output database.

Specify NORESERVE only if you can be certain that no one else could be updating the output database while the COPY operation is in process.

If omitted, RESERVE is used by default.

buffer-option

Specifies either BUFFER, to buffer records written to the output database that improves performance and reduces input/output time, or NOBUFFER, to eliminate buffering.

If omitted, NOBUFFER is used by default.

Note: When using BUFFER, an unsuccessful completion could corrupt the output database.

history-option

Deletes detail history when you copy a database.

Specify either HISTDEL to delete the detail history subfile only, or HISTDELALL to delete the history subfile and all basic history records.

Note: When converting databases, if you specify a value for the BLKSIZE operand in the ADDDS control statement that is smaller than was originally specified for the "from" database, specify a value for history-option.

DELBAN Control Statement

The DELBAN control statement is used to delete a specified banner page member from the database.

Syntax

DELBAN *banner-member-name*

where *banner-member-name* specifies the name of the banner page member you want to delete.

DELETE Control Statement

The DELETE control statement is used to delete a database.

The high-level name of the database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

DELETE

DELPAN Control Statement

The DELPAN control statement is used to delete a specified online panel member from the database.

Syntax

DELPAN *onLine-panel-member-name*

where *online-panel-member-name* specifies the name of the online panel member you want to delete.

DELPRSET Control Statement

The DELPRSET control statement is used to delete a specified printer setup member from the database.

Syntax

DELPRSET *prset-member-name*

where *prset-member-name* specifies the name of the printer setup member you want to delete.

HDELETE Control Statement

The HDELETE control statement is used to delete history data from the database. The high-level name of the database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Important! Before you issue the HDELETE control statement, stop all CA Deliver started tasks that reference the database from which you are deleting history data. Ensure that no reports or bundles are active. Active reports and bundles contain pointers to history file records that are invalid after the HDELETE command is run. The use of these invalid pointers in bundle printing can cause RMODBI11 messages to be issued. The easiest way to ensure that there are no active reports and bundles is to run a MAKECKPT EMPTY control statement before running the HDELETE control statement.

Syntax

HDELETE *detail-option*

where:

detail-option

Specifies whether to delete or clear the records that contain existing history-definition data in the database.

Valid options are as follows:

DETAIL

Deletes the detail records.

This is the default.

ALL

Also deletes the basic history records.

HDELETE ALL

Deletes both basic and detail history records. The space utilized by the basic history records is marked as available for use but the blocks that contained those records are not returned to the database as free blocks. To reclaim these blocks, reorganize the database. This can be accomplished by doing either a RMODBASE COPY into a new database or a RMODBASE UNLOAD/RELOAD into a new database.

LOAD Control Statement

The LOAD control statement is used for recovery purposes to load the database from a sequential, previously unloaded file. You define the input (unloaded) copy of the database with DD statement RMOLOAD.

The LOAD control statement with an UNLOAD control statement can be used to reorganize a database and reclaim blocks in the database that are allocated but are not currently being utilized.

The high-level name of the database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement). The output database must be a newly created, empty database.

If LOAD is attempted against a database that already contains data then a message RMODBM11 TO DATABASE IS NOT EMPTY is issued and the step terminates with a return code 16.

Syntax

LOAD buffer-option reserve-option history-option

where:

buffer-option

Specifies either BUFFER to buffer records written to the output database that improves performance and reduces input/output time or NOBUFFER to eliminate buffering

If omitted, NOBUFFER is used by default.

Important! When using BUFFER, an unsuccessful completion could corrupt the output database.

reserve-option

Specifies either RESERVE, to place a reserve on the output database for the duration of the copy operation, or NORESERVE, to not place a reserve on the output database

Use NORESERVE only when you are certain that no one else could be updating the output database while the LOAD operation is in process.

If omitted, RESERVE is used by default.

history-option

Deletes detail history when you load a database

You can specify either HISTDEL to delete the detail history subfile only or HISTDELALL to delete the history subfile and all basic history records.

Note: When converting databases, if you specify a value for the BLKSIZE operand in the ADDDS control statement that is smaller than was originally specified for the "from" database, specify a value for history-option.

MAKECKPT Control Statement

The MAKECKPT control statement is used to create and recreate the checkpoint.

Syntax for a Non-SMS data set

```
MAKECKPT UNIT=unit VOLSER=volser CYLINDER=cyls empty-opt
```

Syntax for an SMS data set

```
MAKECKPT CYLINDER=cyls DATACLAS=dc MGMTCLAS=mc STORCLAS=sc empty-opt
```

where:

unit

Specifies the unit name to be used to dynamically allocate the new checkpoint.

This operand is optional.

volser

Specifies the volume serial number on which to allocate the new checkpoint.

This operand is required.

cyls

Specifies the number of contiguous cylinders to allocate to the new checkpoint, from 2–65534.

Specify an even number of cylinders. If you specify an odd number, one is added to the value.

This operand is required.

Note: Although you can specify from 2–65534 cylinders for this operand, CA Deliver currently supports a maximum of 40 cylinders.

empty-opt

Specifies whether a new, empty checkpoint is to be created, if one exists.

If you specify EMPTY, a new, empty checkpoint is created.

This operand is optional.

If omitted, data from the old, existing checkpoint (if it exists) is copied to the new checkpoint.

Dc

Specifies the SMS data class you want to assign to the data set for the database.

This operand is optional and works only if you have IBM's SMS.

mc

Specifies the SMS management class you want to assign to the data set for the database.

This operand is optional and works only if you have IBM's SMS.

sc

Specifies the SMS storage class you want to assign to the data set for the database.

This operand is optional and works only if you have IBM's SMS.

Example

```
MAKECKPT UNIT=3380 VOLSER=MVS501 CYLINDER=22
```

Note: For best performance, place the checkpoint on a fast-write, cached device; never place it on the same volume as the CA Deliver or CA View databases.

Interdependency of MAKECKPT Parameters

The following table summarizes the interdependency of the MAKECKPT parameters for SMS and non-SMS data sets. It also lists the parameters that you must use, parameters you can optionally use, and parameters you do not specify.

Parameter	Data Set Not Managed by SMS	Data Set Managed by SMS
UNIT	Optional	Optional
VOLSER	Optional	Optional
CYLINDER	Required	Required
DATACLAS	Do not specify	Optional
MGMTCLAS	Do not specify	Optional
STORCLAS	Do not specify	Optional

Important! When you allocate your database, be sure to allocate enough space for both your data and a copy of half of the allocated cylinders for your checkpoint data set. If you do not allocate enough space, the checkpoint task abnormally ends, and takes CA Deliver down in the process.

Calculating Cylinders for Your Checkpoint

Because the length of a checkpoint (the number of bytes per checkpoint) in the checkpoint data set can vary, you can only determine the number of cylinders to set aside for your checkpoint by trial and error.

On average the checkpoint can hold approximately 5,000 non-bundled report entries or 2,500 bundled report entries per cylinder. Those numbers are further reduced if reports belong to multiple bundles. Active report and bundle entries accumulate in the checkpoint during the daily cycle, so the checkpoint must be allocated large enough to hold a day's worth of active entries.

If, while running CA Deliver, you receive error message RMOCP05 80 PERCENT UTILIZATION OF CHECKPOINT DSN=DATA SET NAME on your console, shut down CA Deliver, increase the number of cylinders allocated for your checkpoint with RMODBASE by 25%, then restart CA Deliver. You can run the RMOCPMAP utility to determine how the checkpoint space is being used.

Note: For best performance, place the checkpoint on a fast-write, cached device.

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database. The NAME control statement applies to all control statements that follow until another NAME control statement is encountered or changed by the RENAME control statement.

If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used.

Syntax

NAME *high-level-name*

where *high-level-name* specifies the high-level name for the database. It is comprised of one or more qualifiers separated by periods. The maximum length of the name is 17 characters. This operand is required.

OLOAD Control Statement

The OLOAD control statement is used to load the panel, message, and skeleton JCL members in the online library to the database. The online library is defined with the DD statement RMOOLIB.

The high-level name of the database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

OLOAD

PLOAD Control Statement

The PLOAD control statement is used to load the printer setup members in the printer setup library to the database. The printer setup library is defined with the DD statement RMOPLIB.

The high-level name of the database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

PLOAD

You can specify a maximum of 50 print lines in a printer setup member.

Note: The PRSET field on the Report Definition Attributes panel, which you use to specify the name of the printer setup member in the database, is described in the *Administration Guide*.

RENAME Control Statement

The RENAME control statement is used to rename a database.

The high-level name of the database to be renamed must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

RENAME *new-name*

where *new-name* specifies the high-level name to which the database is to be renamed. The maximum length of the name is 17 characters. This operand is required.

After completing the rename operation, the high-level name of the database that is used by subsequent operations is set to the new name of the database.

For example, the statement:

```
RENAME DELIVER.NEWSYS1
```

is identical to the following two statements:

```
RENAME DELIVER.NEWSYS1  
NAME DELIVER.NEWSYS1
```


STATUS Control Statement

The STATUS control statement is used to display usage statistics about the CA Deliver database.

These usage statistics are as follows:

- Total blocks
- Total blocks used
- Percentage of utilization
- Cylinders, blocks, used blocks, and error blocks for each data set in the database
- Number of empty blocks (without data) in the database.

The high-level name of the database must have been defined previously with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

STATUS *display-option* *reserve-option*

where

display-option

Specifies FULL to print detailed statistics for the database and data sets or INDEX to print detailed statistics for the database only.

If the parameter is omitted, summary statistics for the database and data sets are printed.

reserve-option

Specifies either RESERVE to place a reserve on the database for the duration of the database evaluation process or NORESERVE to not place a reserve on the database.

This option is only applicable when specified with the *display-option*. Because the database is updated by many tasks, jobs, or online sessions, the database data can change during the evaluation process, causing inconsistent statistics. The RESERVE option delays external updates to the database for the purpose of obtaining consistent statistics. If *reserve-option* is omitted, NORESERVE is assumed.

To compile detailed statistical information, the entire database must be read. Depending on the size of the database, this process can take a considerable amount of time.

Messages Resulting From STATUS

The STATUS statement causes the RMODBA09 messages to be displayed. The descriptions of the database statistics displayed in the RMODBA09 messages are as follows:

Index levels

Specifies the number of levels or hierarchies in the database index.

Index records

Specifies the total number of records in the database.

Index allocations

Specifies the total number of control areas or segments that comprise the database.

Index blocks

Specifies the total number of database blocks that comprise the database index.

These index blocks can be pointer blocks, data blocks, or free blocks.

Index used blocks

Specifies the total number of database blocks currently being used in the database index.

This value includes the index pointer block, the index data blocks, and the index empty blocks.

Index pointer blocks

Specifies the total number of higher level hierarchical index blocks that reference lower level index blocks.

Index data blocks

Specifies the total number of database blocks that comprise the lowest level of the database index, and that contain the physical database data.

Index empty blocks

Specifies the total number of index data blocks that do not contain any index records.

When index records are deleted from the database index, the index data is removed from the index data block. When all index records have been removed from the index data block, the index data block is considered empty but available for subsequent use in the same key range. COPY or UNLOAD/LOAD functions of the RMODBASE utility are recommended for a database with a high percentage of empty blocks.

UNLOAD Control Statement

The UNLOAD control statement is used to unload the database to a sequential, output file for back-up purposes. The records in the database are written in ascending key-sequence order to the RMOUNLD data set as variable length records. The high-level name of the database must have been previously defined with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

The UNLOAD control statement with an LOAD control statement can be used to reorganize the database and reclaim blocks in the database that are allocated but are not currently being utilized.

Syntax

UNLOAD *reserve-option history-option*

where:

reserve-option

Specifies RESERVE to place a reserve on the database for the duration of the unload. If omitted, no reserve is placed on the database.

Note: *Do not* specify NORESERVE; it is not a valid option for the unload.

history-option

Deletes detail history when you unload a database.

You can specify only HISTDELALL to delete the detail history subfile and all basic history.

Note:

- When converting databases, if you specify a value for the BLKSIZE operand in the ADDDS control statement that is smaller than was originally specified for the "from" database, specify a value for *history-option*.
- Back up your CA Deliver data sets regularly. You can recover your database with the LOAD control statement, which automatically reorganizes a database when restoring it. You can use IEBCOPY or any partitioned data set copy utility to back up (and recover) partitioned data sets such as the model banner page library.

VERSION Control Statement

The VERSION control statement is used to version the database to a specific release level. The VERSION control statement can be used to version a Release 11, Release 11.5, Release 11.6, Release 11.7, or Version 12.0 database to Release 12.1 or a Release 12.1 database to Version 12.0, Release 11.7, Release 11.6, Release 11.5, or Release 11. To convert a release 1.6 or a release 1.7 database, use the CONVERT control statement.

The high-level name of the database must have been previously defined with the NAME control statement (or the PARM parameter of the EXEC JCL statement).

Syntax

VERSION *release-number*

where:

release-number specifies the release level to which the database is to be versioned. If omitted, the database is versioned to the current release.

Valid values are 11.0, 11.5, 11.6, 11.7, 12.0, and 12.1

Examples of Control Statements

The following example illustrates how to create a database with a high-level name of DELIVER.SYSTEM1 and allocate 10 cylinders on 3390 volume RMO001.

Note: The high-level name is limited to 17 characters.

```
//EXAMPLE1 JOB ACCOUNT,PROGRAMMER
//STEP1 EXEC PGM=RMOBASE
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NAME DELIVER.SYSTEM1
ADDDS CYLINDER=10 UNIT=3390 VOLSER=RMO001
MAKECKPT CYLINDER=10 UNIT=3390 VOLSER=RMO004 EMPTY
//
```

The database defined in the previous example is out of space. For this part of the example, an additional 40 cylinders are added on 3390 volume RMO002. After space is added, statistics are displayed.

```
//EXAMPLE2 JOB ACCOUNT,PROGRAMMER
//STEP1   EXEC PGM=RMODBASE
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
NAME DELIVER.SYSTEM1
ADDD5 UNIT=3390 VOLSER=RMO002 CYLINDER=40
STATUS
//
```

Changes are made to the online library, which is to be reloaded to DELIVER.SYSTEM1.

```
//EXAMPLE3 JOB ACCOUNT,PROGRAMMER
//STEP1   EXEC PGM=RMODBASE,PARM='DELIVER.SYSTEM1'
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//RMOOLIB DD DSN=CAI.CVDEPENU,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
OLOAD
//
```

The checkpoint data set is to be expanded for existing database DELIVER.SYSTEM1, as shown in the following example.

```
//EXAMPLE4 JOB ACCOUNT,PROGRAMMER
//STEP1   EXEC PGM=RMODBASE
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
NAME DELIVER.SYSTEM1
MAKECKPT UNIT=3390 VOLSER=RMO004 CYLINDER=10
STATUS
//
```

Changes have been made to the model banner page library. It is to be reloaded to database DELIVER.SYSTEM1.

```
//EXAMPLE4 JOB ACCOUNT,PROGRAMMER
//STEP1   EXEC PGM=RMODBASE,PARM='DELIVER.SYSTEM1'
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//RMOBLIB DD DSN=CAI.CVDED133,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
BLOAD
//
```

Database DELIVER.SYSTEM1 has continually grown in size with the addition of many data sets. For performance reasons it is to be copied to one new large database. The name of the old database (DELIVER.OLDSYS1) is to be kept. This sample job is shown as follows:

```
//EXAMPLE5 JOB ACCOUNT,PROGRAMMER
//STEP1 EXEC PGM=RMOBASE
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NAME DELIVER.SYSTEM1
RENAME DELIVER.OLDSYS1
NAME DELIVER.SYSTEM1
ADDDS CYLINDER=60 UNIT=3390 VOLSER=RM0001
MAKECKPT UNIT=3390 VOLSER=RM0004 CYLINDER=10
COPY DELIVER.OLDSYS1
//
```

Database DELIVER.OLDSYS1 is no longer being used and is to be deleted. This sample JCL shows how to delete a database.

```
//EXAMPLE6 JOB ACCOUNT,PROGRAMMER
//STEP1 EXEC PGM=RMOBASE
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NAME DELIVER.OLDSYS1
DELETE
//
```

The database is to be unloaded on a periodic basis for backup purposes; the unloaded backup is to be a generation data set on tape. Sample JCL is shown as follows:

```
//EXAMPLE7 JOB ACCOUNT,PROGRAMMER
//STEP1 EXEC PGM=RMOBASE
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//RMOUNLD DD DSN=DELIVER.BACKUP.RMOBASE(+1),
//          DISP=(,CATLG),UNIT=TAPE
//SYSIN DD *
NAME DELIVER.SYSTEM1
UNLOAD
//
```

The database is to be recovered. It is to be restored from the most recent backup (see the previous example). The database is named DELIVER.SYSTEM1 and is first to be renamed to DELIVER.OLDSYS1. This sample job recovers the database as shown as follows:

```
//EXAMPLE8 JOB ACCOUNT,PROGRAMMER
//STEP1 EXEC PGM=RMOBBB
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//RMOLOAD DD DSN=DELIVER.BACKUP.RMOBBB(+0),DISP=OLD
//SYSIN DD *
NAME DELIVER.SYSTEM1
RENAME DELIVER.OLDSYS1
NAME DELIVER.SYSTEM1
ADDDS CYLINDER=60 UNIT=3390 VOLSER=RMO001
MAKECKPT UNIT=3390 VOLSER=RMO004 CYLINDER=10
LOAD
//
```

The database is to have all history records (including basic history) deleted. The following sample deletes the history records.

```
//EXAMPLE9 JOB ACCOUNT,PROGRAMMER
//STEP1 EXEC PGM=RMOBBB
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NAME DELIVER.SYSTEM1
HDELETE ALL
//
```

RMOBBB samples are in the following CVDEJCL members:

- HBRNADDS
- HBRNBLOD
- HBRNOLOD

RMOBBB - Database Construction from Existing Data

You use the online facility to define distribution identifiers as described in the *Administration Guide*. If your distribution information is in a machine-readable form, you can also use the RMOBBB utility to define and assign distribution identifiers to users and locations.

The RMODBB utility program allows you to initially construct or modify the database from existing data you are currently maintaining about the reports you produce and distribute. The utility allows you to quickly and easily add distribution specifications to the report definitions created by the RMOJCL utility from its scan of your JCL for the job.

The RMODBB utility can also be used to perform batch updating (including adding, changing, deleting, and renaming) of job, report, distribution, and bundle identifiers in the database.

The RMODBB utility reads any sequential file containing fixed, variable, or undefined length records. Control statements define the data fields to be added to or modified in the database and provide the format of the records in the input file.

Important! The started task must be executing on the same system image as all batch and online facilities that access checkpoint data, detail historical data, and facilities you use to delete definitions for these batch and online utilities to work.

Updating Active Definitions

Be aware of the following conditions:

- You can use RMODBB to update definitions while the started task is executing and accessing the database you specified for RMODBB.
- You must *not*, however, update an active definition with RMODBB.
Do not update a definition when that definition is active (displayed as ACTIVE on the Report Panel or on the Active Bundle Panel) or is being used by an executing application job.

When making any updates to the CA Deliver definitions, be sure to test the updates before applying them to a live system. Any incorrect updates could result in reports not being processed and distributed or processed incorrectly.

Job Control Statements

Specify the following JCL to execute RMOBBB:

JOB

Initiates the job.

EXEC

Specifies the utility program name (PGM=RMOBBB) and optionally, the high-level name of the database as the PARM parameter (PARM='DELIVER.SYSTEM1').

STEPLIB DD

Identifies the load library that contains RMOBBB.

If the utility program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

DATA DD

Defines the sequential input data set.

The data set can contain fixed, variable, or undefined length records.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

The keyword TEST can be coded as the second sub parameter of the PARM field in the EXEC control statement to indicate that a test run of the utility is to be made (for example, PARM='DELIVER.SYSTEM1,TEST'). A test run processes all the input data; however, no data is added to the database.

DBASE Control Statement

The DBASE control statement is used to specify the high-level name of the database. The DBASE control statement applies to all control statements following it until another DBASE control statement is encountered.

If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used.

Syntax

`/DBASE NAME=high-level-name`

where *high-level-name* specifies the high-level name for the database; it is composed of one or more qualifiers separated by periods. The maximum length of the name is 17 characters. This operand is required.

Data Definition Control Statements

You can use four control statements to modify the data definitions in the database. These control statements are as follows:

/BNLDEF

Adds, changes, deletes, or renames bundle definitions.

/DISTDEF

Adds, changes, deletes, or renames distribution definitions.

/JOBDEF

Adds, changes, deletes, or renames report specifications for job definitions.

/RPTDEF

Adds, changes, deletes, or renames report definitions.

Syntax

`keyword=(column, length)`

where:

keyword

Specifies the name of the data field.

column

Specifies the beginning column number of the data field in the input records.

length

Specifies the length of the data field in the input records

The length can be omitted to use the default length. If omitted, the parentheses surrounding the column and length can also be omitted, for example, `keyword=column`.

Each data definition control statement processes the complete input data file from the beginning. If necessary, a data record is extended on the right side with blanks to satisfy the parameter specifications.

BNDLDEF Control Statement

The BNDLDEF control statement is used to add, change, delete, and rename bundle definitions.

The parameters and their descriptions and defaults are as follows:

BNDLDEF Keyword	Description	Default Length	Required
BANNER1	Model banner page name for bundle	8	No
BANNER2	Model banner page name for distribution	8	No
BANNER3	Model banner page name for reports	8	No
BCONFIRM	Bundle confirmation indicator The data values are: Y Specifies that a bundle is not to be printed until a user enters the P tabular command on the Active Bundle List panel N Specifies that a user does not need to enter the P tabular command on the Active Bundle List panel to print the bundle Blank Uses the value to which the initialization parameter BNDLCONF is set	1	No
BDISTID	Distribution identifier for distributing the complete bundle	32	No
BID	Bundle identifier	32	Yes
DENTNO	Distribution identifier entry number The data values are: U Identifies the distribution entry (by number) that is to be updated DD Identifies the distribution entry that is to be deleted Blank Identifies where the entry is to be inserted The data value must be from 0–32767. See the topic Using Combinations of DENTNO, DRELNO, DISTID, and ODISTID Parameters later in this section.	3	No
DESC	Bundle description	40	No
DISTID	Optionally specifies a specific (PROD) or generic (PROD*) distribution identifier that is to be added to the end of the distribution specifications for the bundle definition Blank specifies that no distribution identifier is added for the record.	32	No

BNDLDEF Keyword	Description	Default Length	Required
DRELNO	<p>Specifies the distribution identifier relative entry number</p> <p>The data values are as follows:</p> <p>U Identifies the relative occurrence of the distribution identifier (ODISTID or DISTID) that is to be updated</p> <p>DD Identifies the relative occurrence of the distribution identifier (ODISTID or DISTID) that is to be deleted</p> <p>Blank Specifies after which an entry is to be inserted</p> <p>The data value must be from 0–32766.</p> <p>See the topic Using Combinations of DENTNO, DRELNO, DISTID, and ODISTID Parameters later in this section.</p>	3	No
FUNCTION	<p>Processing function</p> <p>The data values are:</p> <p>Blank Adds or changes bundle definitions; this is the default function</p> <p>D Deletes the bundle definition; only the BID parameter is used</p> <p>DD Deletes all distribution specifications, a series of distribution entries, or an individual distribution entry</p> <p> See the topic Using Combinations of DENTNO, DRELNO, DISTID, and ODISTID Parameters later in this section.</p> <p>DI Deletes all special instructions or an individual special instruction entry</p> <p> See the topic Using the IENTNO Parameter later in this section.</p> <p>DR Deletes all report identifiers, a series of report identifiers, or an individual report identifier</p> <p> See the topic Using Combinations of RENTNO, RRELNO, ORID, and RID Parameters later in this section.</p> <p>R Renames the bundle definition; only the OBID and BID parameters are used</p> <p> Specify the original and new bundle identifiers, respectively.</p> <p>SD Sorts the distribution specifications for the bundle in ascending sequence</p>	2	No

BNDLDEF Keyword	Description	Default Length	Required
	SR Sorts the report specifications for the bundle in ascending sequence		
	U Updates bundle definition, bundle report identifiers, bundle distribution identifiers, or bundle special instructions		
IENTNO	Special instruction entry number The data value identifies the special instruction entry by number that is to be updated (function U), deleted (function DI), or where the entry is to be inserted (function blank). The data value must be from 0–32766. See the topic Using the IENTNO Parameter later in this section.	3	No
INST	Special instruction line to be added This parameter is used to add one instruction line to the end of the instructions for the bundle definition. If the data record contains a blank value for INST, no instruction line is added for the record. Any trailing blanks in the data value are truncated from the instruction line.	75	No
	To retain trailing blanks, enclose the instruction line within single quotation marks within the data record. A single quotation mark within the instruction line must be coded as one single quotation mark. Do not duplicate single quotation marks.	75	No
INTERVAL	Bundling interval in a <i>hh:mm</i> format	5	No
JOB CARD <i>n</i>	Job card <i>n</i> , where <i>n</i> is a value from 1–4	72	No
LATE	Late time based on 24-hour clock in a <i>hh:mm</i> format	5	No
OBID	Original bundle identifier for Rename function	32	Yes, for the Rename function
ODISTID	Optionally specifies a specific (PROD) or generic (PROD*) original distribution identifier The data value identifies the distribution identifier that is to be updated (function U), deleted (function DD), or after which an entry is to be inserted (function blank). See the topic Using Combinations of DENTNO, DRELNO, DISTID, and ODISTID Parameters later in this section.	32	No

BNDLDEF Keyword	Description	Default Length	Required
ORID	<p>Original report identifier</p> <p>The data value identifies the report identifier that is to be updated (function U), deleted (function DR), or after which an entry is to be inserted (function blank).</p> <p>See the topic Using Combinations of RENTNO, RRELNO, ORID, and RID Parameters later in this section.</p>	32	
RENTNO	<p>Report identifier entry number</p> <p>The data value identifies the report entry by number that is to be updated (function U), deleted (function DR), or where the entry is to be inserted (function blank).</p> <p>The data value must be from 0–32766.</p> <p>See the topic Using Combinations of RENTNO, RRELNO, ORID, and RID Parameters later in this section.</p>	3	No
RID	<p>Report identifier to be added</p> <p>This parameter is used to add one report identifier to the end of the report specifications for the bundle definition.</p> <p>If the data record contains a blank value for RID, no report identifier is added for the record.</p>	32	No
RRELNO	<p>Report identifier relative entry number</p> <p>The data value identifies the relative occurrence of the report identifier (ORID or RID data value) that is to be updated (function U), deleted (function DR), or after which an entry is to be inserted (function blank).</p> <p>The data value must be from 0–32766.</p> <p>See the topic Using Combinations of RENTNO, RRELNO, ORID, and RID Parameters later in this section.</p>	3	No
WAIT	<p>Wait indicator</p> <p>The data values are Y, N, and blank.</p>	1	No
WAITL	<p>Wait-for-late indicator</p> <p>The data values are Y, N, and blank.</p>	1	No

Rules for Processing Each Data Record (When Adding or Changing)

- If a bundle definition exists for the bundle identifier, that definition is used; otherwise, a new bundle definition is created with all data fields initialized to blanks.
- The data fields in the bundle definition for which parameters were specified on the /BNDLDEF control statement are set to the values from the data record; the data fields for which no parameters were specified on the /BNDLDEF control statement remain unchanged.

DISTDEF Control Statement

The DISTDEF control statement is used to add, change, delete, and rename distribution definitions. The parameters and their meanings and defaults are presented in the following table:

DISTDEF Keyword	Description	Default Length	Required
An	Address line <i>n</i> , where <i>n</i> has a value from 1–9	72	No
CLASS	SYSOUT class	1	No
CONNECT	Indicates whether output is being sent to CA Connect destination. The data values are Y, N, and blank. The default is N.	1	No
DESC	Description for distribution list	40	No
DEST	Destination for printing reports	17	No
DISTID	Distribution identifier	32	Yes
EFORMAT	Email format name	8	No
EMAIL	Email address for DISTID	60	No
FUNCTION	Processing function The data values are: Blank Adds or changes distribution definitions; this is the default D Deletes the distribution definition; only the DISTID parameter is used	2	No

DISTDEF Keyword	Description	Default Length	Required
	<p>DL Deletes all distribution specifications, a series of distribution entries, or an individual distribution entry</p> <p>See the topic Using Combinations of LENTNO, LRELNO, LDISTID, and LODISTID Parameters later in this section.</p> <p>R Renames the distribution definition; only the ODISTID and DISTID parameters are used, and specifies the original and new distribution identifiers</p> <p>SL Sorts distribution specifications for the distribution list in ascending sequence</p> <p>If distribution identifiers are grouped, the group is sorted first and then merged into the list based on the first distribution identifier in the group.</p> <p>U Updates the distribution identifier or distribution list</p>		
L DAYS	<p>Distribution List distribution days for distribution identifier</p> <p>Specify LDISTID with this parameter.</p> <p>This field identifies the days of the week from Monday to Sunday that output is to be produced for the DISTID. A Y or N can be specified for each of these days to indicate whether output is to be produced.</p> <p>The default for the L DAYS field is YYYYYYY.</p> <p>By default, the day that a report is processed by CA Deliver is based on the date and time that the job was submitted and the setting of the BEGINDAY initialization parameter. If the time that the job is submitted is less than the BEGINDAY initialization parameter specification, the day is considered part of the previous day's cycle.</p>	7	No
L DEL	<p>Deletion indicator for distribution identifier in distribution list</p> <p>The data values are Y, N, and blank.</p>	1	No

DISTDEF Keyword	Description	Default Length	Required
LDEST	<p>Destination of printed reports for distribution identifier in distribution list</p> <p>The parameters LDISTID, LDEST, LGROUP, LNDISTID, LWRITER, and LDAYS are used together to add a single distribution specification to the distribution list.</p> <p>If LGROUP is omitted or the data record contains a blank value for LGROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If LGROUP is specified and the data record contains a nonblank value for LGROUP, then the distribution identifier is added to the end of the last group.</p> <p>If the data record contains a blank value for LDISTID, no distribution specification is added for the record.</p> <p>LDEST is only added to the definition, if specified, for the first distribution identifier for a group.</p>	17	No
LDISTID	<p>Distribution identifier to be added to a distribution list</p> <p>To create a distribution list, specify the LDISTID field.</p> <p>The parameters LDISTID, LDEST, LGROUP, LNDISTID, LWRITER, and LDAYS are used together to add a single distribution specification to the distribution list.</p> <p>If LGROUP is omitted or the data record contains a blank value for LGROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If LGROUP is specified and the data record contains a nonblank value for LGROUP, then the distribution identifier is added to the end of the last group.</p> <p>If the data record contains a blank value for LDISTID, no distribution specification is added for the record.</p> <p>LDEST is only added to the definition, if specified, for the first distribution identifier for a group.</p> <p>See the topic Using Combinations of LENTNO, LRELNO, LDISTID, and LODISTID Parameters later in this section.</p>	32	Yes, if LDEL, LDEST, LGROUP, LNDISTID, LWRITER, or LDAYS is specified.

DISTDEF Keyword	Description	Default Length	Required
LENTNO	<p>Distribution identifier entry number in distribution list</p> <p>The data value identifies the distribution entry by number that is to be updated (function U), deleted (function DL), or to locate where the entry is to be inserted (function blank).</p> <p>The data value must be from 0–32767.</p> <p>To specify a larger default length (3), override the field length using the statement.</p> <p>See the topic Using Combinations of LENTNO, LRELNO, LDISTID, and LODISTID Parameters later in this section.</p>	3	No
LGROUP	<p>Grouping indicator for distribution identifier in distribution list</p> <p>The parameters LDISTID, LDEST, LGROUP, LNDISTID, LWRITER, LDAYS are used together to add a single distribution specification to the distribution list.</p> <p>If LGROUP is omitted or the data record is blank for LGROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If LGROUP is specified and the data record contains a nonblank value for LGROUP, then the distribution identifier is added to the end of the last group.</p> <p>If the data record is blank for LDISTID, no distribution specification is added.</p>	1	No
LNDISTID	<p>Number of copies of the report to be produced for distribution identifier in distribution list</p> <p>The parameters LDISTID, LDEST, LGROUP, LNDISTID, and LWRITER are used together to add a single distribution specification to the distribution list.</p> <p>If LGROUP is omitted or the data record contains a blank value for LGROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If LGROUP is specified and the data record contains a nonblank value for LGROUP, then the distribution identifier is added to the end of the last group. If the data record contains a blank value for LDISTID, no distribution specification is added for the record.</p>	3	No

DISTDEF Keyword	Description	Default Length	Required
LODISTID	Original distribution identifier in distribution list Identifies the distribution identifier in the distribution that is to be updated (function U), deleted (function DL), or after which an entry is to be inserted (function blank). See the topic Using Combinations of LENTNO, LRELNO, LDISTID, and LODISTID Parameters later in this section.	32	No
LOUT	Output indicator for distribution identifier in distribution list Specify LDISTID with this parameter. The data values are T (tracking), Y, N, E (Email), and blank.	1	No
LREPT	Reprint indicator for distribution identifier in distribution list The data values are Y, N, and blank.	1	No
LRELNO	Distribution identifier relative entry number in distribution list The data value identifies the relative occurrence of the distribution identifier (LODISTID or LDISTID data value) that is to be updated (function U), deleted (function DL), or after which an entry is to be inserted (function blank). The data value must be from 0–32767. To specify a larger default length (3), override the field length using the statement. See the topic Using Combinations of LENTNO, LRELNO, LDISTID, and LODISTID Parameters later in this section.	3	No
LRVIEW	View restriction indicator for distribution identifier in distribution list The data values are Y, N, and blank.	1	No
LWRITER	External writer name of printed reports for distribution identifier in distribution list The parameters LDISTID, LDEST, LGROUP, LNDISTID, LWRITER, and LDAYS are used together to add a single distribution specification to the distribution list. If LGROUP is omitted or the data record is blank for LGROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group. If LGROUP is specified and the data record contains a nonblank value for LGROUP, then the distribution identifier is added to the end of the last group. If the data record is blank for LDISTID, no distribution specification is added for the record.	8	No

DISTDEF Keyword	Description	Default Length	Required
ODISTID	Original distribution identifier for Rename function	32	Yes, for the Rename function
WEBSVR	Web Server NUMBER (1-4) for DISTID	1	No
WRITER	External writer name	8	No

Rules for Processing Each Data Record (When Adding or Changing)

- If a distribution definition exists for the distribution identifier, that definition is used; otherwise, a new distribution definition is created with all data fields initialized to blanks.
- To create a distribution list, specify the DISTID and LDISTID parameters. The LDISTID, LODISTID, LENTNO, LRELNO, LNDISTID, LDEST, LWRITER, LGROUP, LOUT, LRVIEW, LREPRT, and LDEL keywords are ignored if the distribution identifier is not a distribution list.
- The data fields in the distribution definition for which parameters were specified on the /DISTDEF control statement are set to the values from the data record; the data fields for which no parameters were specified on the /DISTDEF control statement remain unchanged.

JOBDEF Control Statement

The JOBDEF control statement is used to add, change, delete, and rename report specifications for jobs. The parameters and their meanings and defaults are presented in the following table:

JOBDEF Keyword	Description	Default Length	Required
DD	<p>DDname</p> <p>The DD field can be the specific DDname that produces the report in the job or the forms or writer name of the SYSOUT data set being produced. The form and writer name, Sysout Class, or Destination specification is preceded with an "F=", "W=", C=, or D=, respectively. An "*" can also be specified for DD to indicate all SYSOUT data sets.</p> <p>The DD keyword is used with the STEP and PROCSTEP keywords to identify a report being produced in a job.</p> <p>If you want to allow specification of an eight character forms or writer name, override the default length for the DD keyword on the JOBDEF statement to allow for the "F=", "W=", C=, or D= and eight character name (for example, DD=(column,10)).</p>	8	No

JOBDEF Keyword	Description	Default Length	Required
DESC	Job description	40	No
FUNCTION	Processing function The data values are: Blank Adds or changes job definitions This is the default function. D Deletes the job definition and all reports defined for it; only the JOB parameter is used R Renames the job definition; only the OJOB and JOB parameters are used, and specifies the original and new job names, respectively	2	No
INSERT	Report identifier of an existing entry in the job definition after which the new report entry is to be inserted	32	No
JOB	Job name	8	Yes
JPREVRUN	Job-level option Indicates if previously run reports are: D Deleted F Processed unchanged but flagged K Processed and kept unchanged If you omit JPREVRUN and PREVRUN, the value of SPREVRUN is used.	1	No
OJOB	Original job name for the Rename function	8	Yes, for the Rename function
PREVRUN	Report-level option that indicates if previously run reports are: D Deleted F Processed unchanged but flagged K Processed and kept unchanged If you omit PREVRUN, the value of JPREVRUN is used.	1	No
PROCSTEP	Procedure step name	8	No
RID	Report identifier	32	Yes, for the Add/Change function
SPREVRUN	System-level option that indicates if previously run reports are: D Deleted	1	

JOBDEF Keyword	Description	Default Length	Required
	F Processed unchanged but flagged		
	K Processed and kept unchanged		
	If you omit SPREVRUN, JPREVRUN, and PREVRUN, the value K (keep) is used.		No
STEP	Step name	8	No
TYPE	Report type	1	No
	The data values you can specify are as follows:		
	Blank Basic report processing		
	M Monitored data output		
	I Interleaved report processing		
	S Stacked report processing		
	D Dynamic report processing		
	C Control break report processing		

Rules for Processing Each Data Record (When Adding or Changing)

- If a job definition exists for the job name, that definition is used; otherwise, a new job definition is created.
- If the report identifier exists in the job definition, its entry in the job definition is used. The data fields in the entry for which parameters were specified on the /JOBDEF control statement are set to the values from the data record; the data fields for which no parameters were specified on the /JOBDEF control statement remain unchanged.
- If the report identifier exists for a different job definition, an error message is issued and the data record is skipped.
- If the report identifier does not exist, an entry for it is added. If parameter INSERT is specified and exists in the definition, the new entry is inserted immediately after the entry; if parameter INSERT is specified and has a blank value, the new entry is inserted as the first entry in the job definition.
- A report definition is added or modified as necessary.

RPTDEF Control Statement

The RPTDEF control statement is used to add or change report definition attributes. The parameters and their meanings and defaults are presented in the following table:

RPTDEF Keyword	Description	Default Length	Required
ARCH	Archival criteria number The data value must be numeric and from 0–99.	2	No
ARCHID	Archival Report identifier	32	No
BANNER	Model banner page name	8	No
BURST	Burst indicator for IBM 3800 printing. The data values are Y, N, and blank.	1	No
CC	Carriage control identifier	1	No
CCOL	Control break column number The data value must be numeric and from 0–255. A zero or blank value for CCOL, CLEN, or CLINE field indicates that the value is not to be changed. Provide values for all three control break fields for a newly created control report.	3	No
CHAR n	Character arrangement table name n , where n is a value from 1–4	4	No
CLASS	SYSOUT class	1	No
CLEN	Control break field length The data value must be numeric and from 0–255. A zero or blank value for CCOL, CLEN, or CLINE field indicates that the value is not to be changed. Provide values for all three control break fields for a newly created control report.	3	No
CLINE	Control break line number The data value must be numeric and from 0–255. A zero or blank value for CCOL, CLEN, or CLINE field indicates that the value is not to be changed. Provide values for all three control break fields for a newly created control report.	3	No

RPTDEF Keyword	Description	Default Length	Required
CNORM	Control break normalization indicator This indicator determines whether to ignore leading blanks in the control break data. The data values are Y, N, and blank.	3	No
COPIES	Copies indicator The data values are Y, N, and blank.	1	No
COPYG <i>n</i>	IBM 3800 copy group <i>n</i> , where <i>n</i> is a value from 1–8 The data value must be numeric and from 0–255.	3	No
CSEP	Control break separator banner page name	8	No
DAYS	Report distribution days for distribution identifier Specify DISTID with this parameter. This field identifies the days of the week from Monday to Sunday that output is to be produced for the distid. A "Y" or "N" can be specified for each of these days to indicate whether output is to be produced. The default for the DAYS field is "YYYYYYY". By default, the day that a report is processed by CA Deliver is based on the date and time that the job was submitted and the setting of the BEGINDAY initialization parameter. If the time that the job is submitted is less than the BEGINDAY initialization parameter specification, the day is considered part of the previous day's cycle.	7	No
DEL	Deletion indicator The data values are Y, N, and blank.	1	No
DENTNO	Distribution identifier entry number The data value identifies the distribution entry by number that is to be updated (function U), deleted (function DD), or to locate where the entry is to be inserted (function blank). The data value must be from 0–32767. To specify a larger default length (3), override the field length using the statement. If you use DENTNO to delete a REPT type distribution entry, both the REPT and all associated DIST entries are deleted.	3	No
DESC	Report description	40	No

RPTDEF Keyword	Description	Default Length	Required
DEST	<p>Destination for printed reports</p> <p>The parameters DISTID, DEST, GROUP, NDISTID, WRITER, and DAYS are used together to add a single distribution specification to the report definition.</p> <p>If GROUP is omitted or the data record contains a blank value for GROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If GROUP is specified and the data record contains a nonblank value for GROUP, then the distribution identifier is added to the end of the last group.</p> <p>If the data record contains a blank value for DISTID, no distribution specification is added for the record.</p> <p>DEST is only added to the definition, if specified, for the first distribution identifier for a group.</p>	17	No
DISTID	<p>Distribution identifier to be added to the definition</p> <p>The parameters DISTID, DEST, GROUP, NDISTID, WRITER, and DAYS are used together to add a single distribution specification to the report definition.</p> <p>If GROUP is omitted or the data record contains a blank value for GROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If GROUP is specified and the data record contains a nonblank value for GROUP, then the distribution identifier is added to the end of the last group. If the data record contains a blank value for DISTID, no distribution specification is added for the record.</p> <p>DEST is only added to the definition, if specified, for the first distribution identifier for a group.</p> <p>If DTYPE is specified and the data record contains a value of 'R', then the DISTID represents the dynamic report id.</p> <p>If DTYPE is specified and the data record contains a value of 'D', then the DISTID represents the dynamic distribution id.</p>	32	Yes, if DEL, DEST, DTYPE, GROUP, NDISTID, WRITER, or DAYS is specified.

RPTDEF Keyword	Description	Default Length	Required
DRELNO	<p>Distribution identifier relative entry number</p> <p>The data value identifies the relative occurrence of the distribution identifier (ODISTID or DISTID data value) that is to be updated (function U), deleted (function DD), or after which an entry is to be inserted (function blank).</p> <p>The data value must be from 0–32767.</p> <p>To specify a larger default length (3), override the field length using the statement.</p> <p>If you use DRELNO to delete a REPT type distribution entry, both the REPT and all associated DIST entries are deleted.</p>	3	No
DTYPE	<p>Distribution identifier type for dynamic report</p> <p>Specify DISTID with this parameter.</p> <p>The data values are:</p> <p>Blank For non-dynamic reports</p> <p>R Identifies the distribution identifier as a dynamic report name</p> <p>D Indicates the distribution identifier as a dynamic distribution name</p> <p>After a distribution identifier is defined with a DTYPE of R or D it cannot be updated (U) to another DTYPE value. Use a delete (DD) and add function instead of an update (U) when the wrong DTYPE is specified.</p> <p>If DTYPE=R is specified during a delete (DD), only dynamic report ids that match the DISTID are deleted.</p> <p>If DTYPE=D is specified during a delete (DD), only dynamic distribution names that match the DISTID are deleted.</p> <p>If you use DTYPE of R to delete a REPT type distribution entry, both the REPT and all associated DIST entries are deleted.</p> <p>If you use DTYPE of D to delete a DIST entry, and it is the only DIST entry defined to the associated REPT entry, both the DIST and REPT entries are deleted.</p>	1	No
FCB	Forms control image name	4	No
FLASH	IBM 3800 forms flash overlay name	4	Yes, if FLASHCT is specified
FLASHCT	IBM 3800 forms flash count	3	No

RPTDEF Keyword	Description	Default Length	Required
	Data values must be numeric and from 0–255.		
FORM	Special forms name	8	No
FORMDEF	Form definition name for the IBM 3800 printing subsystem	6	No
FUNCTION	Processing function The data values are as follows:	2	No
DD	Deletes all distribution specifications, a series of distribution entries, or an individual distribution entry. For more information about processing function, see the topic Using Combinations of DENTNO, DRELNO, DISTID, DTYPE, and ODISTID Parameters later in this section.		
DI	Deletes all special instructions or an individual special instruction entry See the topic Using the IENTNO Parameter later in this section.		
DT	Deletes all text specifications or an individual text specification entry.		
DV	Deletes all variable specifications or an individual variable entry. Variable specification entries that are currently defined within a dynamic report's distribution specification cannot be deleted.		
D	Deletes the report definition and all references to it; only the RID parameter is used.		
R	Renames the report definition (only the ORID and RID parameters are used), and specifies the original and new report identifiers, respectively		
SD	Sorts distribution specifications for the report in ascending sequence If distribution identifiers are grouped, the group is sorted first and then merged into the list based on the first distribution identifier in the group.		
SV	Sorts variable specifications for the report in ascending sequence.		

RPTDEF Keyword	Description	Default Length	Required
	<p>U Updates the report definition, report distribution data, report text, or report special instruction.</p> <p>Blank Adds or changes report definitions; this is the default</p>		
GROUP	<p>Grouping indicator for distribution identifier</p> <p>The parameters DISTID, DEST, GROUP, NDISTID, WRITER, and DAYS are used together to add a single distribution specification to the report definition.</p> <p>If GROUP is omitted or the data record is blank for GROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If GROUP is specified and the data record contains a nonblank value for GROUP, then the distribution identifier is added to the end of the last group.</p> <p>If the data record is blank for DISTID, no distribution specification is added for the record.</p>	1	No
GROUPID	<p>Name of the output group to be used when the report is printed.</p> <p>If an archival copy of the report is sent to a View FSS collector, this name identifies the member name in the SARINDEX data set which contains archive control statements for the report.</p>	8	No
IENTNO	<p>Special instruction entry number</p> <p>The data value identifies the special instruction entry by number that is to be updated (function U), deleted (function DI), or to locate where the entry is to be inserted (function blank).</p> <p>The data value must be from 0–32766.</p>	3	No
INST	<p>Special instruction line to be added</p> <p>The INST parameter is used to add one instruction line to the end of instructions for the report definition.</p> <p>If the data record is blank for INST, no instruction line is added for the record. Any trailing blanks in the data value are truncated from the instruction line.</p> <p>To retain trailing blanks, enclose the instruction line within single quotation marks within the data record. A single quotation mark within the instruction line must be coded as one single quotation mark. Do not duplicate single quotation marks.</p>	75	No

RPTDEF Keyword	Description	Default Length	Required
LATE	Late time based on a 24-hour clock in a <i>hh:mm</i> format (for example, 14:20)	5	No
MODIFY	Copy modification module name	4	Yes, if TRC is specified
NDISTID	<p>Number of copies of the report to be produced for the distribution identifier</p> <p>The parameters DISTID, DEST, GROUP, NDISTID, WRITER, and DAYS are used together to add a single distribution specification to the report definition.</p> <p>If GROUP is omitted or the data record contains a blank value for GROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If GROUP is specified and the data record contains a nonblank value for GROUP, then the distribution identifier is added to the end of the last group. If the data record contains a blank value for DISTID, no distribution specification is added for the record.</p>	3	No
ODISTID	<p>Original distribution identifier</p> <p>Identifies the distribution identifier that is to be updated (function U), deleted (function DD), or after which an entry is to be inserted (function blank).</p>	32	No
OPTCDJ	<p>OPTCD=J indicator for 3800 printing</p> <p>The data values are Y, N, and blank.</p>	1	No
ORID	Original report identifier for the Rename function	32	Yes, for the Rename function
OUT	<p>CA View output indicator</p> <p>Specify DISTID with this parameter.</p> <p>The data values are T (tracking), Y, N, E (Email), and blank.</p>	1	No
PAGEDEF	Page definition name for the IBM 3800 printing subsystem	6	No
PRMODE	<p>Process mode required to print a SYSOUT data set</p> <p>Specify LINE, PAGE, or a valid mode printer defined specifically for your site.</p>	8	No
PRSET	Printer setup name	8	No

RPTDEF Keyword	Description	Default Length	Required
PRTY	<p>Priority at which a SYSOUT data set enters the output queue</p> <p>Specify a decimal value from 1 (lowest priority) to 255 (highest priority) or a space or 0. If a space or 0 is entered, it is treated as a null and nullifies any existing PRTY value.</p>	3	No
REPRT	<p>Reprint indicator</p> <p>The data values are Y, N, and blank.</p>	1	No
RID	Report identifier	32	Yes
RVGE5	<p>View restriction indicator</p> <p>The data values are Y, N, and blank.</p> <p>The parameter is the same as RVIEW but retained for compatible with older releases.</p>	1	No
RVIEW	<p>View restriction indicator</p> <p>The data values are Y, N, and blank.</p>	1	No
TENTNO	<p>Number that identifies the identification text entry by the number to be updated (function U), deleted (function DT), or to locate where the entry is to be inserted (function blank)</p> <p>Data values must be in the range 0–32766.</p>	3	No
TEXT	<p>Report identification text</p> <p>The parameters TEXT, TEXTCOL, TEXTLINE, TEXTCOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE are used together to add a single report identification text entry to the end of the specifications for the report definition.</p> <p>If the data record contains a blank value for TEXT or a blank value for TEXTLINE, then no identification text specification is added for the record. Any trailing blanks in the data value for TEXT are truncated.</p> <p>To retain trailing blanks, enclose the identification text within single quotation marks within the data record. Leading blanks are retained if they are enclosed in single quotation marks on the input record.</p> <p>A single quotation mark within the identification text must be coded as one single quotation mark. Do not duplicate single quotation marks—the text is always entered "as is"; it is not translated to uppercase.</p>	53	Yes, if either TEXTCOL, TEXTLINE, TEXTCOL, TEXTLIN, TEXTOP, TEXTREUS, or TEXTTYPE is specified

RPTDEF Keyword	Description	Default Length	Required
TEXTCOL	<p>Column number on report page to look for report identification text</p> <p>The data values must be in the range 0–32760 or "*". A zero value or "*" indicates that the entire report line is to be scanned for the report identification text.</p> <p>The parameters TEXT, TEXTCOL, TEXTLINE, TEXTECOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE are used together to add a single report identification text entry to the end of the specifications for the report definition.</p> <p>If the data record contains a blank value for TEXT, then no identification text specification is added for the record.</p> <p>To specify a larger default length (3) for TEXTCOL, override the field length using the statement.</p>	3	No
TEXTLINE	<p>Line number on report page to look for report identification text</p> <p>The data value must be in the range 0–255 or "*". A zero value or "*" indicates that each line of the report page, up to 255 lines, is to be scanned for the report identification text.</p> <p>The parameters TEXT, TEXTCOL, TEXTLINE, TEXTECOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE are used together to add a single report identification text entry to the end of the specifications for the report definition.</p> <p>If the data record contains a blank value for TEXT, then no identification text specification is added for the record.</p>	3	Yes, if TEXT or TEXTCOL is specified
TEXTECOL	<p>Ending Column number on report page to look for report identification text</p> <p>The data values must be in the range of a space or 0-32760. A zero and space indicate that the parameter is not used.</p> <p>The data value must not be less than the specified TEXTCOL value. The TEXTCOL to TEXTECOL range must also be large enough to contain the Report identification text value specified in the TEXT parameter.</p> <p>The parameters TEXT, TEXTCOL, TEXTLINE, TEXTECOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE are used together to add a single report identification text entry to the end of the specifications for the report definition.</p> <p>If the data record contains a blank value for TEXT, then no identification text specification is added for the record.</p> <p>To specify a larger default length (3) for TEXTECOL, override the field length using the statement.</p>	3	No

RPTDEF Keyword	Description	Default Length	Required
TEXTLIN	<p>Ending Line number on report page to look for report identification text</p> <p>The data values must be in the range of a space or 0-255. A zero and space indicate that the parameter is not used.</p> <p>The value must not be less than the specified TEXTLINE value.</p> <p>The parameters TEXT, TEXTCOL, TEXTLINE, TEXTECOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE are used together to add a single report identification text entry to the end of the specifications for the report definition.</p> <p>If the data record contains a blank value for TEXT, then no identification text specification is added for the record.</p>	3	No
TEXTOP	<p>Text comparison operator</p> <p>Text in the report is compared to the text in the database. Specify <i>one</i> of the following values:</p> <p>EQ, blank = equal to</p> <p>NE or ^= not equal to</p> <p>LT or < less than</p> <p>GT or > greater than</p> <p>LE or <= less than or equal to</p> <p>GE or >= greater than or equal to</p> <p>The parameters TEXT, TEXTCOL, TEXTLINE, TEXTECOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE are used together to add a single report identification text entry to the end of the specifications for the report definition.</p>	2	No
TEXTREUS	<p>Segment reusability indicator</p> <p>The data values are Y, N, and blank.</p> <p>The parameters TEXT, TEXTCOL, TEXTLINE, TEXTECOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE are used together to add a single report identification text entry to the end of the specifications for the report definition.</p>	1	No
TEXTTYPE	<p>Type of text entry</p> <p>Specify <i>one</i> of the following values:</p> <p>A, & Continuation test using "and" logic</p> <p>B Beginning test on overlapping segment</p> <p>E Ending test on overlapping segment</p> <p>O, Continuation test using "or" logic at the segment level</p>	1	No

RPTDEF Keyword	Description	Default Length	Required
	P Tests for selecting a page overlapping segment		
	R Continuation test using "or" logic for previous statement		
	X Beginning test on exclusive segment		
	Blank Treat as X for first entry and A for entries other than first		
TRC	IBM 3800 copy modification table reference character	1	No
UCS	Universal character set name	4	No
USCOL n	Control break column number of user field number n , where n equals 1–9 Data values must be numeric and from 0–255. A blank value for the USCOL n , USLEN n , or USLIN n field indicates that the value is not to be changed. Provide values for all three related user fields when initially setting their values. To remove their values, specify a zero value for all three related user fields.	3	No
USLEN n	Control break field length of user field number n , where n equals a value from 1–9 Data values must be numeric, and from 0–255. A blank value for the USCOL n , USLEN n , or USLIN n field indicates that the value is not to be changed. Provide values for all three related user fields when initially setting their values. To remove their values, specify a zero value for all three related user fields.	3	No
USLIN n	Control break line number of user field number n , where n is a value of 1–9 Data values must be numeric and from 0–255. A blank value for the USCOL n , USLEN n , or USLIN n field indicates that the value is not to be changed. Provide values for all three related user fields when initially setting their values. To remove their values, specify a zero value for all three related user fields.	3	No

RPTDEF Keyword	Description	Default Length	Required
VENTNO	<p>Number that identifies the variable specification entry by the number to be updated (function U), deleted (function DV), or to locate where the entry is to be inserted (function blank)</p> <p>Data values must be in the range 0–32766.</p>	3	No
VARNAME	<p>Name used to uniquely identify the variable for this report location. The field consists of 1-8 alphanumeric (A-Z, 0-9) or national (@, #, \$) characters. The first character must be alphabetic or national</p>	8	Yes, if either VARCOL, VARLINE,, or VARLEN is specified
VARCOL	<p>Column number on the page where variable data is to be extracted</p> <p>The data values must be in the range 0–32760. A zero and space indicate that the parameter is not used.</p> <p>The parameters VARNAME, VARCOL, VARLINE and VARLEN are used together to add a single variable specification.</p> <p>Default: 1</p> <p>To specify a larger default length (3) for VARCOL, override the <i>length</i> parameter in VARCOL=(column,length)</p>	3	No
VARLINE	<p>Line number on the page where variable data is to be extracted</p> <p>The data value must be in the range 0–255. A zero and space indicate that the parameter is not used.</p> <p>The parameters VARNAME, VARCOL, VARLINE and VARLEN are used together to add a single variable specification.</p> <p>Default: 1</p>	3	No
VARLEN	<p>Length of the variable data</p> <p>The data values must be in the range of a space or 0-32. A zero and space indicate that the parameter is not used.</p> <p>The parameters VARNAME, VARCOL, VARLINE and VARLEN are used together to add a single variable specification.</p> <p>Default: 1</p> <p>If VARCOL + VARLEN exceed 32760, VARLEN is adjusted to correct value automatically.</p> <p>To specify a larger default length (3) for VARLEN, override the <i>length</i> parameter in VARCOL=(column,length).</p>	3	No

RPTDEF Keyword	Description	Default Length	Required
WRITER	<p>External writer name for printed reports or Email format override for email reports</p> <p>The parameters DISTID, DEST, GROUP, NDISTID, WRITER, and DAYS are used together to add a single distribution specification to the report definition.</p> <p>If GROUP is omitted or the data record is blank for GROUP, then the distribution identifier is added to the end of the distribution specifications as a new, single-identifier group.</p> <p>If GROUP is specified and the data record contains a nonblank value for GROUP, then the distribution identifier is added to the end of the last group.</p> <p>If the data record is blank for DISTID, no distribution specification is added for the record.</p>	8	No

More information:

[Using Combinations of the DENTNO, DRELNO, DISTID, and ODISTID Parameters](#) (see page 157)

[Using the IENTNO Parameter](#) (see page 158)

[Using the TENTNO Parameter](#) (see page 160)

Rules for Processing Each Data Record (When Adding or Changing)

- If a report definition does not exist, an error message is issued and the data record is skipped.
- The data fields in the report definition for which parameters were specified on the /RPTDEF control statement are set to the values from the data record; the data fields for which no parameters were specified on the /RPTDEF control statement remain unchanged.

See the following parameters for variations on this rule:

- DESTID, DEST, GROUP, and NDISTID
- INST
- TEXT, TEXTCOL, TEXTLINE, TEXTCOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE

Example

A sequential file contains distribution specifications and the identifiers of the reports that are to be used for each distribution specification. The name of the file is PROD.DIST.SPECS. The format of the records in the file is as follows:

Position	Field Description
1–32	DISTID
33–52	Address line 1
53–72	Address line 2
73–92	Address line 3
93–124	Report identifier 1
125–156	Report identifier 2
157–188	Report identifier 3
189–220	Report identifier 4
221–242	Report identifier 5
253–254	FUNCTION keyword

As a result, the following job, which is located in RMOBB in CAI.CVDEJCL, is executed:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//DBB EXEC PGM=RMOBB,PARM='DELIVER.SYSTEM1'
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DATA DD DSN=PROD.DIST.SPECS,DISP=SHR
//SYSIN DD *
/DISTDEF DISTID=1 A1=(33,20) A2=(53,20) A3=(73,20)
/RPTDEF RID=93 DISTID=1 FUNCTION=(253,2)
/RPTDEF RID=125 DISTID=1 FUNCTION=(253,2)
/RPTDEF RID=157 DISTID=1 FUNCTION=(253,2)
/RPTDEF RID=189 DISTID=1 FUNCTION=(253,2)
/RPTDEF RID=221 DISTID=1 FUNCTION=(253,2)
/*
```

Using Combinations of Parameters and Functions

You can use combinations of the parameters and functions of the BNDLDEF, DISTDEF, and RPTDEF control statements to modify the data definitions in the database. These combinations and the specific operations they perform are outlined in this section.

Using Combinations of the DENTNO, DRELNO, DISTID, and ODISTID Parameters

Typically you specify *one* of the following:

- ODISTID to change, update, or delete distribution information and DISTID to add distribution information
- ODISTID or DISTID to change, update, or delete specific report IDs within a bundle

In cases when you have defined multiple ODISTIDs or DISTIDs with the same name, use the DRELNO and DENTNO parameters to identify which ODISTIDs or DISTIDs you want to process.

The parameters DENTNO, DRELNO, DISTID, and ODISTID can be used to identify a specific distribution entry or a series of distribution entries that are:

- To be updated (function U)
- To be deleted (function DD)
- To locate where a distribution entry is to be inserted or added (function blank)

The DENTNO and DRELNO parameters are ignored if the data value is zero or blank.

The DENTNO parameter maintains precedence over the other parameters.

Operations Performed Against the Distribution Specifications

The following table describes the operations performed against the distribution specifications for combinations of parameters and functions. Insertion and deletion immediately affect the sequencing of the distribution specifications and also affect the sequencing of subsequent operations.

Parameter	Blank Function	U Function	DD Function
DENTNO	Not applicable; DISTID required for addition	Updates DISTID at entry number	Deletes DISTID at entry number
DENTNO and DISTID	Adds DISTID at entry number	Updates DISTID at entry number	Deletes DISTID at entry number DISTID ignored
DRELNO and ODISTID	Not applicable; DISTID required for addition	Updates relative occurrence of ODISTID	Deletes relative occurrence of ODISTID

Parameter	Blank Function	U Function	DD Function
DRELNO, ODISTID, and DISTID	Adds DISTID after the relative occurrence of ODISTID	Updates/renames relative occurrence of ODISTID; renames if DISTID is not equal to ODISTID	Deletes relative occurrence of ODISTID; DISTID is ignored
DRELNO and DISTID	Adds DISTID after the relative occurrence of DISTID	Updates the relative occurrence of DISTID	Deletes the relative occurrence of DISTID
ODISTID and DISTID	Adds DISTID after the first occurrence of ODISTID	Updates/renames all occurrences of ODISTID; renames if DISTID is not equal to ODISTID	Deletes all occurrences of DISTID
DISTID	Adds DISTID to the end of DISTID specifications	Updates all occurrences of DISTID	Deletes all occurrences of DISTID
DENTNO, DRELNO, ODISTID, and DISTID not specified	Not applicable	Updates first DISTID entry	Deletes all DISTID specifications

Using the IENTNO Parameter

Typically you specify the INST parameter to add special instructions. In some cases, when you want to add or update specific special instructions, use the IENTNO parameter.

The parameter IENTNO can be used to identify a specific special instruction entry that is:

- To be updated (function U)
- To be deleted (function DI)
- To locate where a special instruction entry is to be inserted or added (function blank)

The IENTNO parameter is ignored if the data value is zero or blank.

To specify a larger default length (3) for IENTNO, override the field length using the statement.

Operations Performed Against the Special Instruction Specifications

The following table describes the operations performed against the special instruction specifications for the IENTNO parameter. Insertion and deletion immediately affect the sequencing of the special instruction specifications and affect the sequencing of subsequent operations.

Parameter	Blank function	U Function	DI Function
IENTNO	Adds special instruction line at entry number	Updates special instruction line at entry number	Deletes special instruction line at entry number
IENTNO not specified	Adds special instruction line at the end of special instruction specifications	Updates first special instruction entry	Deletes all special instruction specifications

Using Combinations of the RENTNO, RRELNO, ORID and RID Parameters

Typically you specify ORID or RID to change, update, or delete specific report IDs within a bundle. In cases when you have defined multiple report IDs with the same name, the RRELNO and RENTNO parameters are used to identify which report IDs you want to process.

The parameters RENTNO, RRELNO, RID, and ORID can be used to identify a specific report entry or a series of report entries that are:

- To be updated (function U)
- To be deleted (function DR)
- To locate where a report entry is to be inserted or added (function blank)

The RENTNO and RRELNO parameters are ignored if the data value is zero or blank.

The RENTNO parameter maintains precedence over the other parameters.

Operations Performed Against the Report Specifications

The following table describes the operations performed against the report specifications for combinations of parameters and functions. Insertion and deletion immediately affect the sequencing of the report specifications and also affect the sequencing of subsequent operations.

Parameter	Blank Function	U Function	DR Function
RENTNO	Not applicable; RID required for addition	Updates report at entry number	Deletes report at entry number
RENTNO and RID	Adds RID at entry number	Updates/renames RID at entry number	Deletes report at entry number; RID is ignored
RRELNO and ORID	Not applicable; RID required for addition	Updates/renames relative occurrence of ORID	Deletes relative occurrence of ORID
RRELNO, ORID, and RID	Adds RID after the relative occurrence of ORID	Updates/renames relative occurrence of ORID; renames if RID not equal to ORID	Deletes relative occurrence of ORID; RID is ignored
RRELNO and RID	Adds RID after the relative occurrence of RID	Updates the relative occurrence of RID	Deletes the relative occurrence of RID
ORID and RID	Adds RID after the first occurrence of ORID	Updates/renames all occurrences of ORID; rename if RID not equal to ORID	Deletes all occurrences of RID; RID is ignored
RID	Adds RID to end of report specifications	Updates all occurrences of RID	Deletes all occurrences of RID
RENTNO, RRELNO, ORID, and RID not specified	Not applicable	Updates first report entry	Deletes all report specifications

Using the TENTNO Parameter

Typically you specify an identification text entry according to the instructions for TEXT, TEXTCOL, TEXTLINE, TEXTCOL, TEXTLIN, TEXTOP, TEXTREUS, and TEXTTYPE. TENTNO are used in cases when control at the level of entry numbers is required.

The parameter TENTNO can be used to identify a specific identification text entry that is:

- To be updated (function U)
- To be deleted (function DT)
- To locate where an identification text entry is to be inserted or added (function blank)

The TENTNO parameter is ignored if the data value is zero or blank.

To specify a larger default length (3) for TENTNO, override the field length using the statement.

Operations Performed Against the Identification Text Specifications

The following table describes the operations performed against the identification text specifications for combinations of parameters and functions. Insertion and deletion immediately affect the sequencing of the identification text specifications and also affect the sequencing of subsequent operations.

Parameter	Blank Function	U Function	DT Function
TENTNO	Adds text line at entry number	Updates text line at entry number	Deletes text line at entry number
TENTNO not specified	Adds text line at end of text specifications	Updates first text entry	Deletes all text specifications

Using Combinations of the LENTNO, LRELNO, LDISTID, and LODISTID Parameters

Typically you specify LODISTID to change, update, or delete distribution information and LDISTID to add distribution information in a distribution list.

In cases when you have defined multiple LODISTIDs or LDISTIDs with the same name, the LRELNO and LENTNO parameters are used to identify which LODISTIDs or LDISTIDs you want to process.

The parameters LENTNO, LRELNO, LDISTID, and LODISTID can be used to identify a specific distribution entry or a series of distribution entries that are:

- To be updated (function U)
- To be deleted (function DL)
- To locate where a distribution entry is to be inserted or added (function blank)

The LENTNO and LRELNO parameters are ignored if the data value is zero or blank.

The LENTNO parameter maintains precedence over the other parameters.

Operations Performed Against the Distribution List Specifications

The following table describes the operations performed against the distribution list distribution specifications for combinations of parameters and functions. Insertion and deletion immediately affect the sequencing of the distribution specifications and affect the sequencing of subsequent operations.

Parameter	Blank Function	U Function	DL Function
LENTNO	Not applicable; LDISTID required for addition	Updates LDISTID at entry number	Deletes LDISTID at entry number
LENTNO and LDISTID	Adds LDISTID at entry number	Updates LDISTID at entry number	Deletes LDISTID at entry number LDISTID ignored
LRELNO and LODISTID	Not applicable; LDISTID required for addition	Updates relative occurrence of LODISTID	Deletes relative occurrence of LODISTID
LRELNO, LODISTID, and LDISTID	Adds LDISTID after the relative occurrence of LODISTID	Updates/renames relative occurrence of LODISTID; renames if LDISTID is not equal to ODISTID	Deletes relative occurrence of LODISTID; LDISTID is ignored
LRELNO and LDISTID	Adds LDISTID after the relative occurrence of LODISTID	Updates the relative occurrence of LDISTID	Deletes the relative occurrence of LDISTID
LODISTID and LDISTID	Adds LDISTID after the first occurrence of LODISTID	Updates/renames all occurrences of LODISTID; renames if LDISTID is not equal to LODISTID	Deletes all occurrences of LDISTID
LDISTID	Adds LDISTID to the end of LDISTID specifications	Updates all occurrences of LDISTID	Deletes all occurrences of LDISTID
LENTNO, LRELNO, LODISTID, and LDISTID not specified	Not applicable	Updates first LDISTID entry	Deletes all LDISTID specifications

RMODSC Report Data Set Collector

In addition to the pre-spool and post-spool collection interfaces, CA Deliver provides an interface to collect, distribute, and bundle report data contained in a standard sequential file. By default, CA Deliver distributes reports based on lookup of job name, step name procedure step name, and data definition (DD) name information. The data set collector provides added flexibility by allowing multiple files to be processed, allowing distribution based on a specific report definition, and allowing manipulation and/or sectioning of input data through a named user exit.

A sequential input file containing the report data is designated by either the PARM field or control statement specifications. All standard record formats, record lengths, and block sizes are supported. A user exit can also be designated to add, change, and/or delete input records and specifically identify sections of the report data.

Data Set Collector Considerations

Certain requirements and restrictions govern the use of the data set collector which are:

- The CA Deliver started task for the specified database must be started and running on the same system as the data set collector job.
- Non-bundled report output is produced under the job and step of the job that executes the data set collector utility.
- No distribution output is generated for monitor output (the monitored report can however be archived and/or bundled).
- UNDEF reports are not generated for non-matched job name, step name, procedure step name, data definition name, and/or report name (UNDEF reports can however be generated for non-matched stacked report pages).
- If a report definition name is designated through control statement specification or user exit determination, you need not define the report under the execution job name. The step name, procedure step name, and DDname defined under the job are also ignored but are defined to differentiate it from other reports.
- The data set collector does not carry forward any default JCL attributes for the reports. If specific attributes are required for the report, define them in the CA Deliver database within the report or distribution specifications.

Job Control Statements

Specify the following JCL to execute RMODSC:

JCL

Description.

JOB

Initiates the job.

EXEC

Specifies the name of the program (PGM=RMODSC) and optional parameters (PARM=) for the program.

Use the optional PARM parameter to specify the high-level name of a database, input DD statement name, and user exit name in the following format:

PARM='HIGH-LEVEL-NAME,DDNAME,EXITNAME'

These parameters can also be specified on the NAME and COLLECT control statement.

If the SYSIN DD statement is not specified, the PARM field must specify the high-level name of the database and optionally the input DD name. If the input DD name is not specified, a DD name of REPORT is used.

STEPLIB DD

Identifies the load library that contains RMODSC and required user exits.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

input DD

Specifies a sequential input file containing the report data to be collected by CA Deliver.

The input file must be a physical sequential data set (DSORG=PS) that contains fixed, variable, or undefined records. Define the input file with machine or ASA carriage control characters for stacked or control report processing.

The name of the DD statement that is to be collected is designated in the PARM field and/or on the COLLECT control statement.

NAME Control Statement

The NAME control statement identifies the high-level name of the database and optionally the name of the user exit to process the input file.

Syntax is as follows:

```
NAME HIGH-LEVEL-NAME EXIT-NAME
```

where:

high-level-name

Defines an active CA Deliver started task for this database that must be running on the same system as the job that is executing the RMODSC program.

exit-name

Optionally specifies the name of the user exit to be used during the processing of the input file

If omitted, the input file is processed without a user exit.

COLLECT Control Statement

The COLLECT control statement identifies the data definition (DD) name of the input file to be collected by CA Deliver and optionally the name of the report definition used to distribute the report.

Syntax

```
COLLECT ddname report-name
```

where:

ddname

Specifies the data definition (DD) name of the sequential input file to be collected by CA Deliver

The input file must be a physical sequential data set (DSORG=PS) that contains fixed, variable, or undefined records. Define the input file with machine or ASA carriage control characters for stacked or control report processing.

report-name

Optionally specifies the name of the report definition that is to be used for distributing the report. If the report-name contains an imbedded space or comma, enclose it in quotes – single (') or double (").

If report-name is specified, you need not define that report under the execution job name. The step name, procedure step name, and DDname defined under the job is also ignored but are defined to differentiate it from other reports. The report can be defined as a basic, stacked, or control report. For stacked and control reports, this report designates the first report in a series of stacked or control reports.

If omitted, CA Deliver performs a database lookup based on the execution job name, step name, procedure step name, and this ddname to determine how to distribute the report.

Input User Exit

See the RMODSCUX description in the User Exits section of the *Programming Guide*.

Example #1

The following example shows sample JCL to collect and distribute the report data contained in data set DELIVER.REPORT1.DATA. A basic report has been defined to CA Deliver under job RMORPT1 with a step name of REPORT and a DDNAME of INFILE.

```
//RMORPT1 JOB ACCOUNT,PROGRAMMER,CLASS=A,MSGCLASS=A
//REPORT EXEC PGM=RMODSC,PARM='DELIVER.SYSTEM1,INFILE'
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//INFILE DD DSN=DELIVER.REPORT1.DATA,DISP=SHR
```

Example #2

The following example shows sample JCL to collect and distribute the report data contained in data set DELIVER.REPORT2.DATA. The data set is distributed as report SALESRPT-R1. The SALESRPT-R1 is defined as a stacked report which allows further breakdown of the report.

```
//RMORPT2 JOB ACCOUNT,PROGRAMMER,CLASS=A,MSGCLASS=A
//REPORT EXEC PGM=RMODSC
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SALESRPT DD DSN=DELIVER.REPORT2.DATA,DISP=SHR
//SYSIN DD *
NAME DELIVER.SYSTEM1
COLLECT SALESRPT SALESRPT-R1
/*
```

Example #3

The following example shows sample JCL to collect and distribute the report data contained in data set DELIVER.Z1FILE.DATA. This example uses a user exit called RMODSCU1 (the exit is in CVDEOPTN) to scan position 1 and 2 of each report line for the characters 'Z1'. When the Z1 line is found, it starts a new report whose name is found in positions 16 to 25 of the Z1 record. In addition, the Z1 record is replaced with a record that contain '1' in column 1 for the ASA carriage control character and 'REPORT ID=' followed by the report name column 9.

```
//RMRPT3 JOB ACCOUNT,PROGRAMMER,CLASS=A,MSGCLASS=A
//REPORT EXEC PGM=RMODSC,
//          PARM='DELIVER.SYSTEM1,Z1FILE,RMODSCU1'
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//Z1FILE DD DSN=DELIVER.Z1FILE.DATA,DISP=SHR
```

RMOGRW - General Report Writer

The RMOGRW program is a general purpose reporting utility that provides hardcopy printout or data set output of information in the database. The utility can obtain any information from the following database records:

- Job descriptor record
- Report descriptor record
- Report history record
- Report detail history record
- Distribution data record
- Bundle descriptor record
- Bundle history record
- Bundle detail history record
- Active report status record
- Active bundle status record

The database records are not referenced explicitly, but implicitly using field names. The utility accesses the appropriate database records depending on the field names that are referenced. A full list of field names is provided later in this chapter.

The utility provides free-format control statements to sort, print, output, compare, and select fields from the database. In addition, statements are provided to define report titles, to define special fields, and to manipulate special fields. These special fields can also be used for sorting, printing, and other purposes.

Important! CA Deliver's started task must be executing on the same operating system as all batch and online facilities that access checkpoint data detail, historical data, and facilities you use to delete definitions for these batch and online facilities to work.

Job Control Statements

Specify the following JCL to execute RMOGRW:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMOGRW) and optionally, the high-level name of the database as the PARM parameter (PARM='DELIVER.SYSTEM1').

You may also need to specify a region size (REGION=4096K is recommended).

STEPLIB DD

Identifies the load library that contains RMOGRW.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

PRTFILE DD

Defines the sequential output data set (typically SYSOUT) to which the hardcopy report is to be written using the PRINT control statement.

The DD statement name can be changed by use of the CONTROL control statement.

OUTFILE DD

Defines the sequential output data set to which records are to be output using the OUTPUT control statement.

The DD statement name can be changed by use of the CONTROL control statement. The DCB parameters LRECL, BLKSIZE, and RECFM are required to avoid a system error.

SORTLIB DD

Defines the load library containing the sort library programs.

SORTWK01 DD

Defines sort work disk space.

SORTWK02 DD

Defines sort work disk space.

SORTWK03 DD

Defines sort work disk space.

SYSOUT DD

Defines the sort message data set.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

Control Statements

Specify the following control statements to execute RMOGRW:

BREAK

Discontinues processing of a repetitive group of statements (DO control statement) and proceeds with the statement following the repetitive group of statements.

CONTINUE

Performs the next iteration of a repetitive group of statements (DO control statement).

CONTROL

Specifies alternate values for line count, line size, database high-level name, print file DDname, output filename, and database selection sequence.

DEFINE

Defines the fields to be used to store data or values.

DO

Specifies the beginning of a repetitive group of statements.

ELSE

Specifies the statements that are to be executed when a false condition is determined for an IF control statement.

END

Specifies the end of a repetitive group of statements (DO control statement), the end of a conditional operation (IF control statement), the end of at-end logic (ON control statement), or the end of the control statements.

IF

Specifies the beginning of a conditional operation.

NEXT

Retrieves the next occurrence of a field. You can use this statement only for address and special instruction lines, text entries (line number, column number, text data), history entries, report identifiers, and distribution identifiers.

ON

Specifies the statements that are to be executed when no more database records or sort records are available.

OUTPUT

Specifies the writing of data to the output data set.

PRINT

Specifies the writing of data to the print data set.

RELEASE

Specifies that a sort record is to be constructed and sorted.

SELECT

Specifies a condition or restriction to selection of information from the database.

SET

Specifies the setting of a defined field (DEFINE control statement).

SORT

Specifies a specific sort sequence to be used in ordering information written to the print and output files.

STOP

Specifies the end of a control statement processing phase.

THEN

Specifies the statements that are to be executed when a true condition is determined for an IF control statement.

TITLE

Specifies the definition of a report title.

Field Names

Field names are used to reference and maintain information or data by the general report writer:

- Database field references information from the database.

Database fields are presented chronologically and by database record.

- Defined field saves information independent from the database.
These fields are defined by the DEFINE control statement.
- Reserved field contains the current date, current time, line size, column position, and so on.

Each field name can contain a single type of data. The types of data that can be maintained are as follows:

- Binary (B)
- Character (C)
- Packed (P)
- Date (D)

Dates are maintained as number of days from January 1, 1900 in a binary format.

Julian dates are carried as 4-byte, packed fields defined as *OCYYDDDF*.

where:

<i>C</i>	Represents the century as follows:
0	Represents years 1990 through 1999
1	Represents years 2000 through 2099
<i>YY</i>	Represents the last two digits of the year
<i>DDD</i>	Represents the day of the year (000 to 366)
<i>F</i>	Represents the 4-bit sign character

Example

The Julian date *x'0101025C'* represents January 25, 2001.

Sequence in Which Database Records Are Accessed

The RMOGRW utility program accesses database records based on the database fields that are referenced. The following guidelines determine the sequence in which database records are accessed:

- If database fields from the job descriptor record are referenced, database records are retrieved in job name (JOB) sequence.
- If database fields from the report descriptor record are referenced other than report identifier (RID), database records are retrieved in report identifier (RID) sequence.

- If database fields from the bundle descriptor record or bundle active status record are referenced without referencing database fields from the job descriptor record, report descriptor record, or report active status record other than report identifier (RID), database records are retrieved in bundle identifier (BID) sequence.
- If database fields are referenced from the distribution data record only, database records are retrieved in distribution identifier (DID) sequence.

Because the database fields that are referenced do not necessarily imply the true intent of your request, the sequence in which database records are accessed can be overridden by the CONTROL control statement.

Database Fields for Job Descriptor Record

The following table describes the database fields for the job descriptor record:

Field Name	Description	Length	Type
DD	DD name of the report The DD name field can contain a specific DD statement name, forms name, writer name, or "*". The form, writer name, Sysout class, and Destination specification is preceded with "F=" "W=", C=, or D= respectively.	10	C
JDATE	Date the job was last updated online	4	D
JDESC	Description for the job identifier	40	C
JOB	Job name	8	C
JOBNAME	Same as JOB	8	C
JPREVRUN	Job-level option that indicates if previously run reports are: D Deleted F Processed unchanged but flagged K Processed unchanged If you omit JPREVRUN and PREVRUN, the value of SPREVRUN is used.	1	C
JTIME	Time when the job was last updated online	4	P
JUSER	User to make the last update to job online	8	C
NRID	Number of report identifiers for the job or bundle	2	B
PREVRUN	Report-level option that indicates if previously run reports are: D Deleted	1	C

Field Name	Description	Length	Type
	F Processed unchanged but flagged		
	K Processed unchanged		
	If you omit PREVRUN, the value of JPREVRUN is used.		
PROCSTEP	Procedure step name for the report	8	C
RID	Report identifier	32	C
SPREVRUN	System-level option that indicates if previously run reports are:	1	C
	D Deleted		
	F Processed unchanged but flagged		
	K Processed unchanged		
	If you omit SPREVRUN, JPREVRUN, and PREVRUN, the value K (keep) is used.		
STEP	Step name of the report	8	C

Database Fields for Report Descriptor Record

The following table describes the database fields for the report descriptor record:

Field Name	Description	Length	Type
ARCH	Archival criteria number	2	C
ARCHID	Archival Report identifier	32	C
BANNER	Report banner page name	8	C
BID	Bundle identifier	32	C
BURST	Burst indicator for 3800 printing	1	C
CC	Carriage control identifier	1	C
CCOL	Control break column number	1	B
CHARS	Report control statement entry; print set used	4	C
CHARS n	Character arrangement table name for 3800 printing, where n can range from 1–4	4	C
CLASS	SYSOUT class of the report	1	C
CLEN	Control break field length	1	B
CLINE	Control break line number	1	B

Field Name	Description	Length	Type
CNORM	Control break normalization indicator Valid values are: Y Suppresses leading blanks in the control break data Blank Does not suppress leading blanks in control break data	1	C
COPIES	Copies indicator	1	C
COPYGn	Copy group for IBM 3800 printing; <i>n</i> can be a number from 1–8	1	B
CSEP	Control break separator banner page name	8	C
DAYS	Report distribution days for the distribution identifier This field represents the days of the week from Monday to Sunday that output is to be produced for the distid. A Y or N is entered for each of these days to indicate whether output is to be produced.	7	C
DCOPIES	Number of copies of the report for distribution identifier	1	B
DEST	Print destination for the report	17	C
DID	Distribution identifier	32	C
DISTID	Same as DID	32	C
DTYPE	Distribution identifier (DISTID) type for dynamic report Valid values are: Blank For non-dynamic reports D Distribution identifier is a dynamic distribution name R Distribution identifier is a dynamic report name After a distribution identifier is defined with a DTYPE of R or D it cannot be updated (U) to another DTYPE value. Use a delete (DD) and add function instead of an update (U) when the wrong DTYPE is specified.	1	C
FCB	Form control image name	4	C
FLASH	Forms flash overlay name for IBM 3800 printing	4	C
FLASHCT	Forms flash count for IBM 3800 printing	1	B
FORM	Special forms name	8	C
FORMDEF	Form definition for IBM 3800 printing	6	C

Field Name	Description	Length	Type
GROUP	Grouping indicator for distribution identifier Valid values are: * Groups the distribution identifier Blank Does not group the distribution identifier	1	C
GROUPID	Output group to be used when the report is printed	8	C
INST	Special instruction line for the report	75	C
INSTLEN	Length of the report instruction line (INST)	1	B
INSTLEN <i>n</i>	Length of the report instruction line (INST <i>n</i>)	1	B
INST <i>n</i>	Special instruction line <i>n</i> for the report	75	C
LATE	Report late time	5	C
MODIFY	Copy modification module name	4	C
NBID	Number of bundle identifiers for the report	2	B
NDISTID	Number of distribution identifiers for the report or bundle	2	B
NINST	Number of special instruction lines for the report	2	B
NTEXT	Number of text entries for the report	2	B
NVAR	Number of Dynamic Variables	2	B
OPTCDJ	OPTCD=J indicator for IBM 3800 printing	1	C
OUT	Output indicator for distribution identifier Y Printed output is to be created for the recipient N No printed output or tracking data is to be created T No printed output is to be created, but tracked data is to be created E Email is sent with no printed output	1	C
PAGEDEF	Page definition for an IBM 3800 printer	6	C
PRMODE	Process mode required to print a SYSOUT data set The data values are LINE, PAGE, or a valid process mode defined specifically for your site.	8	C
PRSET	Printer setup name	8	C
PRTY	Output priority at which a SYSOUT data set enters the output queue	1	B
RDATE	Date report was last updated online	4	D

Field Name	Description	Length	Type
RDD	DD statement name for the report The DD name field can contain a specific DD statement name, a forms name, writer name, or "*". The forms, writer, class, and dest specification is preceded with "F=" "W=" "C=" or "D=" respectively.	10	C
RDEL	Restricted deletion indicator Y The user associated with the distribution identifier is allowed to delete the report from CA View N The user associated with the distribution identifier is not allowed to delete the report from CA View	1	C
RDESC	Description for report identifier	40	C
REPR	Reprint indicator Y The user associated with the distribution identifier can reprint the report in CA View N The user associated with the distribution identifier cannot reprint the report in CA View	1	C
RID	Report identifier	32	C
RJOB	Name of job that created the report	8	C
RPROCSTEP	Procedure step name for the report	8	C
RSTEP	Name of step that created the report	8	C
RTIME	Time when the report was last updated online	4	P
RTYPE	Type of report: C Control report M Monitored report processing I Interleaved report processing S Stacked report processing D Dynamic report processing Blank Basic report processing	1	C
RUSER	User to make the last update of the report online	8	C
RVGE5	View restriction indicator The parameter is the same as RVIEW but is retained for compatibility with older releases.	1	C
RVIEW	View restriction indicator for distribution identifier	1	C

Field Name	Description	Length	Type
	Y The user associated with the distribution identifier can only view the report through a non-secured logical view and cannot use the VIEW command		
	N The user associated with the distribution identifier can access any view of the report in CA View		
TEXT	Report identification text	72	C
TEXTCOL	Column number on report page to look for report identification text	2	B
TEXTCOL n	Specific entry for TEXTCOL	2	B
TEXTECOL	Ending Column number on report page to look for report identification text	2	B
TEXTECOL n	Specific entry for TEXTECOL	2	B
TEXTLEN	Length of the report text line (TEXT field)	1	B
TEXTLEN n	Length of the report text line (TEXT n field)	1	B
TEXTLINE	Line number on the report page to look for report identification text	1	B
TEXTLINE n	Specific entry for TEXTLINE	1	B
TEXTLINE	Ending Line number on the report page to look for report identification text	1	B
TEXTLINE n	Specific entry for TEXTLINE	1	B
TEXT n	Specific report identification text line	72	C
TEXTOP	Text comparison operator The report identification text is compared against the report page data based on the comparison operator to determine qualifying pages for the report.	2	C
	EQ Equal to		
	NE Not equal to		
	LT Less than		
	GT Greater than		
	LE Less than or equal to		
	GE Greater than or equal to		
TEXTREUSE	Reusable indicator for report identification text	1	C
	Y Reusable		

Field Name	Description	Length	Type
	Blank Not reusable		
TEXTTYPE	Type of report identification text	1	C
	A Logical "and" continuation test		
	B Beginning test for overlapping segment		
	E Ending test for overlapping segment		
	O Logical "or" continuation test at segment level		
	P Test for selecting a page overlapping segment		
	R Logical "or" continuation test for previous statement		
	X Beginning test for exclusive segment		
TEXTTYPE <i>n</i>	Specific entry for TEXTTYPE	1	C
TRC	Copy modification table reference character for printing on an IBM 3800 printer	1	B
TYPE	Report processing type	1	C
	Blank Basic report processing		
	C Control report processing		
	M Monitored report processing		
	I Interleaved report processing		
	D Dynamic report processing		
	S Stacked report processing		
UCS	Universal character set name	4	C
USCOL <i>n</i>	Control break column number of the user field	1	B
USLEN <i>n</i>	Control break field length of the user field	1	B
USLINE <i>n</i>	Control break line number of the user field	1	B
VARCOL	Column number on report page to look for dynamic variable name	2	B
VARCOL <i>n</i>	Specific entry for VARCOL	2	B
VARLEN	Length of the dynamic variable field	1	B
VARLEN <i>n</i>	Length of the dynamic variable field	1	B
VARLINE	Line number on the report page to look for dynamic variable name	1	B

Field Name	Description	Length	Type
VARLINE <i>n</i>	Specific entry for VARLINE	1	B
VARNAME	Dynamic variable name	8	C
VARNAME <i>n</i>	Specific entry for VARNAME	8	C
WRITER	External writer name for distribution identifier	8	C

Database Fields for Report History Record

The following table describes the database fields for the report history record:

Field Name	Description	Length	Type
HDATE	Date when the report was queued	4	D
HDATE <i>n</i>	Specific history entry for HDATE	4	D
HGEN	Relative generation of the report	4	C
HGEN <i>n</i>	Specific history entry for HGEN	4	C
HJID	Job identifier that produced the report	8	C
HJID <i>n</i>	Specific history entry for HJID	8	C
HJOB	Name of the job that produced the report	8	C
HJOB <i>n</i>	Specific history entry for HJOB	8	C
HLINES	Number of lines queued for the report	4	B
HLINES <i>n</i>	Specific history entry for HLINES	4	B
HPAGES	Number of pages queued for the report	4	B
HPAGES <i>n</i>	Specific history entry for HPAGES	4	B
HPREVRUN	Indicates whether a previously run job was deleted, not rerun, processed unchanged, or processed unchanged but flagged Valid values are: Blank Not rerun D Deleted K Processed unchanged F Processed unchanged but flagged	1	C
HPREVRUN <i>n</i>	Specific history entry for PREVRUN	1	C
HRBLK	Report history original history file block number	4	B

Field Name	Description	Length	Type								
HRBLK <i>n</i>	Specific history entry for HBLK	4	B								
HRDATE	Date when the report was produced	4	D								
HRDATE <i>n</i>	Specific history entry for HDATE	4	D								
HRERUN	Run counter for rerun processing. This counter indicates an initial run, a RUNTARC rerun, or a CA 11 rerun. Valid values are: <table><tr><td>0</td><td>CA 11 Rerun</td></tr><tr><td>1</td><td>Initial run</td></tr><tr><td>>1</td><td>RUNTRAC rerun</td></tr></table>	0	CA 11 Rerun	1	Initial run	>1	RUNTRAC rerun	1	C		
0	CA 11 Rerun										
1	Initial run										
>1	RUNTRAC rerun										
HRERUN <i>n</i>	Specific history entry for HRERUN	1	C								
HRGEN	Relative generation of the report	4	C								
HRGEN <i>n</i>	Specific history entry for HGEN	4	C								
HRJID	Job number of the report	8	C								
HRJID <i>n</i>	Specific history entry for HJID	8	C								
HRJOB	Job name for the report	8	C								
HRJOB <i>n</i>	Specific history entry for HJOB	8	C								
HRLINES	Lines printed for the report	4	B								
HRLINES <i>n</i>	Specific history entry for HRLINES	4	B								
HRPAGES	Pages printed for the report	4	B								
HRPAGES <i>n</i>	Specific history entry for HPAGES	4	B								
HRPREVRUN	Indicates whether a previously run job was deleted, not rerun, processed unchanged, or processed unchanged but flagged. Valid values are: <table><tr><td>Blank</td><td>Not rerun</td></tr><tr><td>D</td><td>Deleted</td></tr><tr><td>K</td><td>Processed unchanged</td></tr><tr><td>F</td><td>Processed unchanged but flagged</td></tr></table>	Blank	Not rerun	D	Deleted	K	Processed unchanged	F	Processed unchanged but flagged	1	C
Blank	Not rerun										
D	Deleted										
K	Processed unchanged										
F	Processed unchanged but flagged										
HRPREVRUN <i>n</i>	Specific history entry for PREVRUN	1	C								
HRRERUN	Run counter for rerun processing. This counter indicates an initial run, a RUNTARC rerun, or a CA 11 rerun. Valid values are: <table><tr><td>0</td><td>CA 11 Rerun</td></tr></table>	0	CA 11 Rerun	1	C						
0	CA 11 Rerun										

Field Name	Description	Length	Type
	1 Initial run		
	>1 RUNTRAC rerun		
HRRERUN n	Specific history entry for HRRERUN	1	C
HRTIME	Time when the report was produced	4	P
HRTIME n	Specific history entry for HTIME	4	P
HTIME	Time when the report was queued	4	P
HTIME n	Specific history entry for HTIME	4	P
NRHIST	Number of history entries for the report	2	B
RID	Report identifier	32	C

Database Fields for Report Detail History Record

The following table describes the database fields for the report detail history record:

Field Name	Description	Length	Type
HDRBHDN	Unique bundle history detail number for a report	10	C
HDRBID	Bundle identifier of bundle where the report resides	32	C
HDRBSN	Bundle sequence number	2	B
HDRDATE	Date when the report was printed	4	D
HDRDDN	DDname used for the produced report	8	C
HDRDID	Distribution identifier for the report	32	C
HDRHDN	History detail number for the report	10	C
HDRJID	Job identifier that produced the report	8	C
HDRJOB	Job name that produced the report	8	C
HDRLINES	Number of lines printed	4	B
HDRPAGES	Number of pages printed	4	B
HDRPDT1	Date posted at station 1	4	D
HDRPDT2	Date posted at station 2	4	D
HDRPDT3	Date posted at station 3	4	D
HDRPDT4	Date posted at station 4	4	D
HDRPDT5	Date posted at station 5	4	D

Field Name	Description	Length	Type
HDRPID	Identifier of the printer that produced the report	8	C
HDRPTM1	Time when posted at station 1	4	P
HDRPTM2	Time when posted at station 2	4	P
HDRPTM3	Time when posted at station 3	4	P
HDRPTM4	Time when posted at station 4	4	P
HDRPTM5	Time when posted at station 5	4	P
HDRRID	Report identifier	32	C
HDRSTA1	Name of station 1	8	C
HDRSTA2	Name of station 2	8	C
HDRSTA3	Name of station 3	8	C
HDRSTA4	Name of station 4	8	C
HDRSTA5	Name of station 5	8	C
HDRTIME	Time when the report is printed	4	P
HDRUSR1	User ID for station 1	12	C
HDRUSR2	User ID for station 2	12	C
HDRUSR3	User ID for station 3	12	C
HDRUSR4	User ID for station 4	12	C
HDRUSR5	User ID for station 5	12	C

Database Fields for Distribution Data Record

The following table describes the database fields for the distribution data record:

Field Name	Description	Length	Type
A	Address line for distribution identifier	72	C
An	Address line <i>n</i> for distribution identifier	72	C
DCLASS	SYSOUT class for distribution identifier	1	C
DCONNECT	CA Connect destination indicator:	1	C
	Y Printed output is being sent to a CA Connect destination		

Field Name	Description	Length	Type
	N Printed output is not being sent to a CA Connect destination		
DDATE	Date the distribution identifier was last updated online	4	D
DDEST	Print destination for distribution identifier	17	C
DID	Distribution identifier	32	C
DISTID	Same as DID	32	C
DLIST	Indicates whether the distribution identifier (DID) is a distribution list	1	C
	Y Distribution identifier is a distribution list		
	N Distribution identifier is not a distribution list		
DTIME	Time when the distribution identifier was last updated online	4	P
DUSER	User to make the last update of the distribution identifier online	8	C
DWRITER	External writer name	8	C
EFORMAT	Email format name	8	C
EMAIL	Email address	60	C
LA	Address line for distribution identifier in distribution list	72	C
LAn	Address line <i>n</i> for distribution identifier in distribution list	72	C
LCOPIES	Number of copies of the report for distribution identifier in distribution list	1	B
LDAYS	Distribution List distribution days for distribution identifier This field represents the days of the week from Monday to Sunday that output is to be produced for the distid. A "Y" or "N" is supplied for each of these days to indicate whether output is to be produced.	7	C
LDEL	Restricted delete indicator for distribution identifier in distribution list	1	C
	Y User associated with the distribution identifier is allowed to delete the report from CA View		
	N User associated with the distribution identifier is not allowed to delete the report from CA View		
LDESC	Descriptor for distribution list The A field name can also be used to extract the description for a distribution list	40	C
LDEST	Print destination for distribution identifier in distribution list	17	C

Field Name	Description	Length	Type
LDISTID	Distribution identifier in distribution list	32	C
LGROUP	Grouping indicator for distribution identifier in distribution list	1	C
	* The distribution identifier is grouped		
	Blank The distribution identifier is not grouped		
LNDISTID	Number of distribution identifiers in distribution list	4	B
LOUT	Output indicator for distribution identifier in distribution list	1	C
	Y Printed output is to be created for the recipient		
	N No printed output or tracking data is to be created		
	T No printed output is to be created, but tracked data is to be created		
	E Email, copy from the OUT field		
LREPT	Reprint indicator for distribution identifier in distribution list	1	C
	Y User associated with the distribution identifier can reprint the report in CA View		
	N The user associated with the distribution identifier cannot reprint the report in CA View		
LRVIEW	View restriction indicator for distribution identifier in distribution list	1	C
	Y The user associated with the distribution identifier can only view the report through a non-secured logical view and cannot use the VIEW command		
	N User associated with the distribution identifier can access any view of the report in CA View		
LWRITER	External writer name for distribution identifier in distribution list	8	C
NA	Number of address lines for distribution identifier	2	B
NLA	Number of address lines for distribution identifier in distribution list	2	B
WEBSVR	WEBSVR Number	1	C

Database Fields for Bundle Descriptor Record

The following table describes the database fields for the bundle descriptor record:

Field Name	Description	Length	Type
BA	Bundle address line for bundle distribution (BDIST)	72	C
BA n	Bundle address line n for bundle distribution (BDIST)	72	C
BBANNER1	Bundle banner page name	8	C
BBANNER2	Bundle distribution banner page name	8	C
BBANNER3	Bundle report banner page name	8	C
BCLASS	Bundle class for bundle distribution (BDIST)	1	C
BCONFIRM	Bundle confirmation indicator:	1	C
	Y That the bundle requires confirmation to print		
	N That the bundle does not require confirmation to print		
	Blank That initialization parameter BNDLCONF is used		
BDATE	Date the bundle was last updated online	4	D
BDESC	Description for bundle identifier	40	C
BDEST	Bundle destination for bundle distribution (BDIST)	17	C
BDIST	Bundle distribution identifier	32	C
BID	Bundle identifier	32	C
BINST	Special instruction line for bundle	75	C
BINSTLEN	Length of bundle instruction line (BINST field)	1	B
BINSTLEN n	Length of bundle instruction line (BINST n field)	1	B
BINST n	Special instruction line n for bundle	75	C
BINTERVAL	Bundle interval time	5	C
BINVL	Same as BINTERVAL	5	C
BJCD	Bundle job card	72	C
BJCD n	Bundle job card where n can be a number from 1–4.	72	C
BLATE	Late time for the bundle	5	C

Field Name	Description	Length	Type
BTIME	Time when bundle was last updated online	4	P
BUSER	User identifier to make last update to bundle	8	C
BWAIT	Bundling wait for interval indicator: Y Wait for interval to bundle N Do not wait for interval to bundle Blank Use default value	1	C
BWAITL	Bundling wait for late time indicator: Y The bundle does not print until the late time has expired N The bundle need not wait for the late time to expire if all reports it needs are available for bundling Blank Use default value	1	C
DID	Distribution identifier	32	C
DISTID	Same as DID	32	C
NBA	Number of bundle address lines	2	B
NBINST	Number of bundle instruction lines	2	B
NDISTID	Number of distribution identifiers for the report or bundle	2	B
NRID	Number of report identifiers for the job or bundle	2	B
RID	Report identifier	32	C

Database Fields for Bundle History Record

The following table describes the database fields for the bundle history record:

Field Name	Description	Length	Type
BID	Bundle identifier	32	C
HBBLK	Bundle history block number	4	B
HBBLK n	Specific history entry for HBBLK	4	B
HBDATE	Date when bundle was queued	4	D
HBDATE n	Specific history entry for HBDATE	4	D
HBGEN	Relative generation of bundle	4	C

HBGEN <i>n</i>	Specific history entry for HBGEN	4	C
HBJID	Job number of the job that produced the bundle	8	C
HBJID <i>n</i>	Specific history entry for HBJID	8	C
HBJOB	Job name of the job that produced the bundle	8	C
HBJOB <i>n</i>	Specific history entry for HBJOB	8	C
HBLINES	Number of lines queued for the bundle	4	B
HBLINES <i>n</i>	Specific history entry for HBLINES	4	B
HBPAGES	Number of pages queued for the bundle	4	B
HBPAGES <i>n</i>	Specific history entry for HBPAGES	4	B
HBSEQ	Bundle history entry sequence number	2	B
HBSEQ <i>n</i>	Specific history entry for HBSEQ	2	B
HBTIME	Time when bundle was queued	4	P
HBTIME <i>n</i>	Specific history entry for HBTIME	4	P
NBHIST	Number of history entries for bundle	2	B

Database Fields for Bundle Detail History Record

The following table describes the database fields for the bundle detail history record:

Field Name	Description	Length	Type
HDBBID	Bundle identifier	32	C
HDBBSN	Bundle sequence number	2	B
HDBDATE	Date when bundle was printed	4	D
HDBDDN	DDname used to print bundle	8	C
HDBDID	Distribution identifier for bundle	32	C
HDBHDN	History detail number for bundle	10	C
HDBJID	Job identifier of the job that printed the bundle	8	C
HDBJOB	Name of the job that printed the bundle	8	C
HDBLINES	Number of lines printed	4	B
HDBPAGES	Number of pages printed	4	B
HDBPDT1	Date posted at station 1	4	D

Field Name	Description	Length	Type
HDBPDT2	Date posted at station 2	4	D
HDBPDT3	Date posted at station 3	4	D
HDBPDT4	Date posted at station 4	4	D
HDBPDT5	Date posted at station 5	4	D
HDBPID	Identifier of the printer that printed the bundle	8	C
HDBPTM1	Time when bundle was posted at station 1	4	P
HDBPTM2	Time when bundle was posted at station 2	4	P
HDBPTM3	Time when bundle was posted at station 3	4	P
HDBPTM4	Time when bundle was posted at station 4	4	P
HDBPTM5	Time when bundle was posted at station 5	4	P
HDBRHDN	History detail number of bundled report	10	C
HDBRID	Report identifier contained in bundle	32	C
HDBSTA1	Name of station 1	8	C
HDBSTA2	Name of station 2	8	C
HDBSTA3	Name of station 3	8	C
HDBSTA4	Name of station 4	8	C
HDBSTA5	Name of station 5	8	C
HDBTIME	Time when the bundle printed	4	P
HDBUSR1	User identification for station 1	12	C
HDBUSR2	User identification for station 2	12	C
HDBUSR3	User identification for station 3	12	C
HDBUSR4	User identification for station 4	12	C
HDBUSR5	User identification for station 5	12	C

Database Fields for Active Report Status Record

The following table describes the database fields for the active report status record:

Field Name	Description	Length	Type
ABID	Bundle identifier for active report entry	32	C

Field Name	Description	Length	Type
ACTIVE	Active report indicator: Y Report is activated Blank Report is not activated	1	C
ADAY	Distribution day for the report This field represents the day of the week that the report was processed (for example, MON, TUE.). By default, the day that a report is processed by Deliver is based on the date and time that the job was submitted and the setting of the BEGIN DAY initialization parameter. If the time that the job is submitted is less than the BEGIN DAY initialization parameter specification, the day is considered part of the previous day's cycle.	3	C
AJOB	Job name for active report entry	8	C
CPUID	CPU that report is being produced on	4	C
EXCP	Exception indicator for the report: Blank No exception N Report was not produced L Report is late	1	C
JOBID	Job number	8	C
RID	Report identifier	32	C
STAT	Status of the report: Blank Report has not been processed yet B Report is awaiting bundling O Report is open Q Report is queued for print P Report is purged	1	C

Database Fields for Active Bundle Status Record

The following table describes the database fields for the active bundle status record:

Field Name	Description	Length	Type
BACTCNT	Indicates the number of reports activated for the bundle	4	B

Field Name	Description	Length	Type
BCPUID	Central processing unit on which the bundle is being produced	4	C
BEXCP	Bundle exception indicator: Blank No exception A Abend N NPROD I Bundle is incomplete	1	C
BID	Bundle identifier	32	C
BPNDCNT	Indicates the number of reports pending activation for the bundle	4	B
BRDYCNT	Indicates the number of reports that are ready to be bundled	4	B
BRINTERVAL	Indicates the remaining interval time in minutes and seconds at which time a bundle can be produced	5	C
BRINVL	Same as BRINTERVAL	5	C
BRPTCNT	Indicates the number of reports that have been bundled	4	B
BSTAT	Bundle status: Blank Bundle has not been processed O Bundle is open Q Bundle is queued for print P Bundle print requested	1	C
BSUB	Bundle submission indicator: Y Bundle batch job has been submitted Blank Bundle batch job has not been submitted	1	C
WTR#	External writer name of report being bundled	6	C

Reserved Fields

The following table describes reserved fields:

Field Name	Description	Length	Type
CDATE	The current date	4	D
COL	The current column number of the printed report being processed	4	B

CTIME	The current time	4	P
DATABASE	Database high-level name	17	C
LINECNT	Printable line per report page that has been processed	4	B
LINESIZE	Width of the report in characters	4	B
OUTFILE	DD statement name of the output file	8	C
PRTFILE	DD statement name of the print file	8	C

Expression

An *expression* can be a field name, a constant, a function, or an operation, as described in the following table:

Field Name	Description
Field name	A database field name, defined field name, or a reserved field name
Constant	A data value that has an unchanging, self-defined value For instance, 1 and ABC are constant values that are not altered during the execution of the statements.
Function	<p>A special routine that is set up to convert, extract, or manipulate data values</p> <p>Functions you can use are as follows:</p> <p>BINARY(<i>expression</i>) Converts the specified expression to internal binary format, resulting in a signed 4-byte binary number</p> <p>CHAR(<i>expression</i>) Converts the specified expression to character format</p> <p>DATE(<i>expression</i>) Converts the specified expression to internal date format. If <i>expression</i> is a character field, the character string must be formatted in the default date format with a date separator character (typically a slash) between the month, day, and year. If expression is a binary or packed field, the field specifies a Julian date with the century indicator in form, CYYDDD.</p> <p>EDIT(<i>exp-1</i>,<i>exp-2</i>) <i>exp-1</i> specifies the expression whose value is to be edited <i>exp-2</i> specifies the edit-pattern to be used for editing <i>exp-1</i>; <i>exp-2</i> is a constant (for example, 'ZZZZ9')</p> <p>The edit-pattern character for a character field is as follows:</p>

Field Name	Description
	<p>X Specifies substitution of a character from <i>exp</i></p> <p>Edit-pattern characters for a numeric field (binary or packed) are as follows:</p> <p>\$ Specifies the substitution of a currency symbol or a floating currency symbol</p> <p>* Specifies that an asterisk or a rolling asterisk is to be substituted in place of a leading zero</p>
Function (Continued)	<p>– Specifies the substitution of a sign or floating sign for a negative value, blank for a positive value. + – specifies the substitution of a sign or floating sign. + is substituted for a positive and - for a negative value</p> <p>.</p> <p>Specifies the substitution of a decimal point</p> <p>,</p> <p>Specifies the substitution of a comma</p> <p>The substitution and suppression of this character can be affected by floating edit characters and zero suppression edit characters.</p> <p>Edit-pattern characters for date fields are as follows:</p> <p>X Specifies the substitution of a numeric character</p> <p>Z Specifies the substitution of a numeric character; if the corresponding digit is a leading zero, blank is substituted</p> <p>9 Specifies the substitution of a numeric character</p> <p>D Specifies the substitution of a one-digit day</p> <p>DD Specifies the substitution of a two-digit day</p> <p>DDD Specifies the substitution of a Julian day</p> <p>M Specifies the substitution of a one-digit month</p> <p>MM Specifies the substitution of a two-digit month</p> <p>Y Specifies the substitution of a one-digit year</p> <p>YY Specifies the substitution of a two-digit year</p> <p>YYY Specifies the substitution of a three-digit year</p> <p>The three-digit year is the number of year from 1900. Year 2000 is represented as 100.</p> <p>YYYY Specifies the substitution of a full four-digit year</p> <p>Z Specifies the substitution of one character from data in the form <i>MMDDYYYY YYDDDD</i> (Gregorian and Julian)</p>

Field Name	Description		
	9	Specifies the substitution of one digit from data in the form MMDDYYYY YYDDD (Gregorian and Julian)	
Function (Continued)	Editing examples:		
	Sending Field	Edit-pattern	Resulting Field
	'ABCDEF'	X-X-X	A-B-C
	'A2534RW'	XXX/XXX/XXX	A25/34R/W
	'BDF'	AXCXEX	ABCDEF
	0	9999	0000
	0	ZZZZ	
	123	ZZ,ZZ9	123
	395	**,**9	***395
	960	\$\$,\$\$9	\$960
	1005	ZZ,ZZ9	1,005
	1256	-----9	1256
	3471	+++++9	+3471
	-523	--,--9	-523
	-7000	++,++9	-7,000
	-9275	ZZ,ZZ9	9,275-
	24569	Z,ZZZ.99+	245.69+
	192543	XX:XX:XX	19:25:43
	999999	ZZZ9	****
	36554 (date)	MM/DD/YYYY	01/30/2000
	36583 (date)	YYYY.MM.DD	2000.02.28
	36645 (date)	DD/MM/YY.DDD	30/04/00.121
	36671 (date)	99/99/9999 999.999	05/26/2000 100.147

Field Name	Description
Function (Continued)	<p>PACK(<i>exp</i>)</p> <p>Converts the specified expression to internal packed format. The resulting value is an 8-byte signed, packed number.</p> <p>PREV(<i>field-name</i>)</p> <p>Specifies the usage of the previous data value for <i>field-name</i>.</p> <p>SUBSTR(<i>exp,pos,len</i>)</p> <p>Specifies the usage of a subset of the specified expression. The SUBSTR expression must result in or be a character value. <i>pos</i> specifies the beginning character to be extracted. <i>len</i> specifies the number of characters to be extracted.</p> <p>TRANS(<i>field,expression-value,expression-result</i> [<i>expression-value, expression-result,...</i>])</p> <p>Specifies the matching of two values (<i>expression</i> and <i>expression-value</i>) and the use of a translated result (<i>expression-result</i>).</p> <p>Expression-value and expression-result can be repeated for each required value and result. An asterisk can be specified as expression-value to indicate any value. If a matching value is not found after interrogation of all values, expression is used as the translated result.</p>
Operation	<p>A group of field names, constants, or functions separated by operators:</p> <p><i>field/constant/function op field/constant/function</i></p> <p>The valid operators (<i>op</i>) are:</p> <ul style="list-style-type: none"> + for adding fields – for subtracting fields * for multiplying fields / for dividing fields for concatenating fields <p>The evaluation and computation of an operation is from left to right in the following order:</p> <ol style="list-style-type: none"> 1. Concatenation 2. Multiplication and division 3. Addition and subtraction <p>The order of evaluation and computation can be altered by enclosing a portion or portions of an operation in parentheses. These parenthetical sections are evaluated first and follow the same order of evaluation as the total operation.</p>

Condition

A *condition* is a group of expressions separated by comparison operators and logical operators. In other words, a condition is used to compare two or more sets of expressions.

Syntax

expression-1 co expression-2 [lo expression-3 co expression-4 ...]

where:

expression-1, *expression-2*, and, optionally, *expression-3* and *expression-4* specify an expression whose value is compared. In addition, *expression-2* and *expression-4* can contain a list of expressions (for example, ('ABC','DEF',...)). Enclose the list of expressions in parentheses.

co is a comparison operator as follows:

=	equal
EQ	equal
^=	not equal
<>	not equal
><	not equal
NE	not equal
<	less than
LT	less than
>	greater than
GT	greater than
^<	greater than or equal to (not less than)
<^	greater than or equal to (not less than)
>=	greater than or equal to
=>	greater than or equal to
GE	greater than or equal to
^>	less than or equal to (not greater than)
>^	less than or equal to (not greater than)
<=	less than or equal to
=<	less than or equal to
LE	less than or equal to

lo is a logical operator as follows:

AND	Conditions on both sides of the logical operator must be true
&	Same as AND
OR	Either condition surrounding the logical operator can be true
	Same as OR

A condition is broken down into groups separated by the OR logical operator if there is any. If this group or any one of these groups is evaluated as true, the total condition is evaluated the same. The process of grouping can be altered by enclosing a portion or portions of the condition in parentheses.

Examples

Check if job name is equal to P25135:

```
JOB = 'P25135'
```

Check if report identifier begins with an R, S, or T:

```
SUBSTR(RID,1,1) = ('R','S','T')
```

Check if history lines are between 5000 and 10000 lines (inclusive):

```
HLINES => 5000 AND HLLINES <= 10000
```

BREAK Control Statement

The BREAK control statement is used with the DO control statement to stop iteration and processing of the DO group where the BREAK control statement is embedded. Processing continues with the control statement following the DO group.

Syntax

```
BREAK
```

The BREAK control statement contains no additional parameters.

CONTINUE Control Statement

The CONTINUE control statement is used with the DO control statement to perform the next iteration and processing of the DO group where the CONTINUE control statement is embedded. If the scope or bounds of the DO operation are reached, processing continues with the control statement following the DO group.

Syntax

```
CONTINUE
```

CONTROL Control Statement

The CONTROL control statement is used to alter the default specification for line count, line size, database ddname, print file ddname, and output file name.

Syntax

CONTROL

LINECNT=*n*
LINESIZE=*n*
DATABASE=*high-level-name*
PRTFILE=*name*
OUTFILE=*name*
ACTIVE=*status*
SEQ=*sequence*
SELECT=*selection*

where:

LINECNT=*n*

Specifies the maximum lines printed per page of output.

n must be a number greater than 10.

The default line count is 60 lines. Aliases of LINECNT are LINECOUNT, LINES, LC, and L.

LINESIZE=*n*

Specifies the maximum width of a print line including the carriage control *n*.

DATABASE=

high-level-name.

Specifies the high-level name of the database

If omitted, the high-level name specified in the PARM parameter on the EXEC JCL statement is used. Aliases for DATABASE are DB and D.

PRTFILE=*name*

Specifies the name of the DD statement to which the PRINT control statement writes its output.

The default ddname is PRTFILE. Aliases for PRTFILE are PRT, PF, and P.

OUTFILE=*name*

Specifies the name of the DD statement to which the OUTPUT control statement writes its output.

The default ddname is OUTFILE. Aliases for OUTFILE are OUT, OF, and O.

ACTIVE=*status*

Specifies whether to use the current or previous active status data when reporting.

status can be specified as C or CURRENT for the current active status data or P or PREV for the previous generation active status data.

SEQ=*sequence*

Specifies the sequence in which database records are selected from the database.

sequence can be specified as JOB for job sequence, RID for report sequence, BID for bundle sequence, or DID or DISTID for the distribution identifier sequence. DID (or DISTID) is only allowed when database fields from the distribution data record are referenced.

SELECT=selection

Identifies a specific selection of identifiers to retrieve from the database.

x can be specified as a specific identifier (PROD123), a generic identifier (PROD*), or a range of identifiers (A:B). A list of specific identifiers, generic identifiers, or ranges of identifiers can be specified by enclosing the list in parentheses.

Example

To specify a line width of 80, a line count of 55, and a print DD statement of PRINT, you enter the following:

```
/CONTROL LINESIZE=80 LINECNT=55 PRTFILE=PRINT
```

DEFINE Control Statement

The DEFINE control statement is used to define fields to retain or store the content of data or values.

Syntax

```
DEFINE field    BIN(len)  
              CHAR(len)  
              DATE  
              PACK(len)
```

Alternate specifications are specified in a columnar fashion.

Important! Do not use the same name as any field in the database or unpredictable results occur.

where:

field

Specifies the name of the field to be defined.

The name can be 1–12 characters in length and must begin with an alphabetic or national character (\$, #, @). Multiple field names can be defined by enclosing a list of field names in parentheses.

BIN(*len*)

Specifies that the field is to be defined as a binary field.

len specifies the length of the binary field.

The length is specified in bytes and can be a number from 1–4. Binary fields other than 4 bytes do not carry a sign. The alias for BIN is BINARY.

If the length is omitted, the field is 4 bytes in length.

CHAR(*len*)

Specifies that the field is to be defined as a character field.

len specifies the length.

The length, specified in bytes, can be a number from 1–256. The alias for CHAR is CHARACTER.

If the length is omitted, the field is 1 byte in length.

DATE

Specifies the definition of a date field.

A date field is maintained internally as the number of days from January 1, 1900 in binary. Date fields print (PRINT control statement) in the default date format and output (OUTPUT control statement) as a 4-byte packed Julian date. These formats can be altered by using the EDIT function.

PACK(*len*)

Specifies that the field is to be defined as a packed field.

len specifies the length of the packed field.

The length is specified in bytes and can be a number from 1–8. The aliases for PACK are PACKED, DEC, and DECIMAL. The default length for the field type (8 bytes) is used if the length is omitted. Defined fields are not sent to or received from sort unless the field is part of the SORT control statement.

Examples

To define a 20-byte character field named CHR, specify:

```
/DEFINE CHR CHAR(20)
```

To define three fields I, J, and K as binary, specify:

```
/DEFINE (I,J,K) BIN
```

DO Control Statement

The DO control statement is used to repeat a given set of statements a specified number of times until a certain condition is met or while a certain condition is met. The set of statements following the DO control statement and preceding the corresponding END control statement constitute a DO group. This DO group continues to execute as long as all necessary conditions are met. As a point of reference, the function and evaluation of the DO control statement itself is referred to as DO operation.

Syntax

```
DO      field = expression-1 TO expression-2 BY expression-3
        FOREVER UNTIL(condition) WHILE(condition)
```

where:

field

Specifies the name of the defined field (DEFINE control statement) that is to be set and incremented during the DO operation.

expression-1

Specifies an expression that defines the initial value for the DO operation.

expression-2

Specifies an expression that defines the final value or limit for the DO operation.

When the final value is exceeded, execution of the DO group is ended.

expression-3

Specifies an expression that defines the incrementing value for the DO operation.

If BY increment-expression is not specified, a value of +1 is used for an ascending range (expression-1 is less than or equal to expression-2), and a value of -1 is used for a descending range (expression-1 is greater than expression-2).

FOREVER

Specifies continual execution of the DO group.

The FOREVER keyword is mutually exclusive with the field = expression-1 TO expression-2 BY expression-3 specification.

UNTIL (condition)

Specifies that the execution of the DO group is to continue until the given condition is true.

WHILE (condition)

Specifies that the execution of the DO group is to continue while the specified condition is true.

All operands for the DO control statement are optional. If no operands are specified, or only field = *expression-1* is specified, the DO group is executed one time only. The DO group continues to be executed until **one** of the following conditions is met:

- The value for field exceeds the range specified by *expression-1* to *expression-2*.

- A true *condition* is received for the UNTIL *condition*.
- A false *condition* is received for the WHILE *condition*.

The execution of the DO group can also be interrupted by a BREAK control statement, by a STOP control statement, or by an end-of-data condition retrieving database records or sort records.

DO control statements can be nested within IF control statements or other DO control statements to any level. Avoid an infinite loop when coding the UNTIL or WHILE conditions.

The DO control statement is convenient for printing multiple occurrences of address lines, special instruction lines, text specification entries, and history entries. The following example shows the printing of multiple occurrences of the text specification line, column, and text.

Example

```
/DEFINE I BIN
/DO I = 1 TO NTEXT BY 1
/  PRINT TEXTLINE  COL(3)
/  PRINT TEXTCOL
/  PRINT TEXT
/  NEXT TEXT
/END
```

ELSE Control Statement

The ELSE control statement is used with the IF control statement to indicate the statements that receives control when a false condition is determined on the IF control statement.

Syntax

```
ELSE
```

END Control Statement

The END control statement is used to specify the end of a DO control statement, an IF control statement, or an ON control statement. There must be a one-to-one correspondence between each DO/IF and END control statement. An END control statement at the lowest level (level 1) indicates the last control statement.

Syntax

```
END
```

IF Control Statement

The IF control statement can be used for conditional execution of statements.

The evaluation of the IF control statement determines a true or false condition. If the condition is true, processing continues with the next statement. If the condition is false, processing continues with the statement following the corresponding ELSE control statement or END control statement if an ELSE control statement was not provided. Conclude the group of statements related to the IF control statement with END.

Syntax

```
IF condition THEN
...
...
ELSE
...
...
END
```

where:

condition

Specifies the condition that is to be checked to determine the sequence of statements to be executed.

ELSE

Used to nest the IF control statement to any level within DO or other IF control statements.

The ELSE control statement is optional.

Examples

To print a more explanatory description for the database field EXCP, specify:

```
/IF      EXCP = ' '
/        PRINT ' '
/        END
/IF EXCP = 'N'
/        PRINT 'NOT PRODUCED'
/        END
/IF EXCP = 'L'
/        PRINT 'LATE'
/        END
```

A more efficient but less readable method is:

```
/IF      EXCP = ' '  
/        PRINT ' '  
/ELSE  
/        IF EXCP = 'N'  
/        PRINT 'NOT PRODUCED '  
/        ELSE  
/            IF EXCP = 'L'  
/                PRINT 'LATE '  
/            END  
/        END  
/END
```

NEXT Control Statement

The NEXT control statement is used to retrieve the next value of a database field. This statement can be used only for retrieving address lines, special instruction lines, text specification entries, history entries, report identifiers, and distribution identifiers.

Syntax

```
NEXT field [, field, ...]
```

where *field* specifies the name of the field whose next value is to be retrieved.

The valid field names that can be referenced on the statement are A, BA, BID, BINST, BHIST, DISTID, HIST, INST, RECORD, RID, TEXT, and VARNAME.

Note: The RECORD field allows you to obtain the next record from the database or the next sort record from the database.

When the NEXT control statement is issued for a field that has no more occurrences, the related field is set to zeros or blanks depending on the field type. RID and DISTID are mutually exclusive with the SORT control statement.

An example of the NEXT control statement is provided in the explanation of the DO control statement.

ON Control Statement

The ON control statement is used to designate a series of statements to be processed when no database records or sort records are available for processing.

The ON control statement can be supplied only once for non-sort logic (no SORT control statement), presort logic (statements preceding the SORT control statement), and logic after the sort (statements following the SORT control statement). The END control statement terminates the series of statements designated by the ON control statement.

Syntax

```
ON ENDDATA
...
END
```

Example

The ON control statement is used for various purposes, but it is especially convenient for printing final totals, as follows:

```
/DEFINE BCNT BIN
/PRINT BID 'BUNDLE, IDENTIFIER'
/SET BCNT=BCNT+1
/ON ENDDATA
/      PRINT 'TOTAL BUNDLES =' SKIP(2)
/      PRINT BCNT
/END
```

OUTPUT Control Statement

The OUTPUT control statement is used to write data to the output file.

Syntax

```
OUTPUT [expression-1] [COL(expression)] [SKIP]
```

where:

expression-1

Specifies an expression that determines the data to be written to the output file.

COL(expression)

Specifies an expression that determines the position to which expression-1 is written.

A value of 1 represents the first position of the output record. If this value is less than the current column position, the current output record is written to the output file, and expression-1 is placed in the new output record.

SKIP

Causes the current output record to be written to the output file.

The data from each OUTPUT control statement is queued contiguously in the output record. The output record is written under the following conditions:

- Statement processing has completed processing for a given database record, and the next OUTPUT control statement does not specify the COL operand.
- The COL operand is specified with a value less than the current column position.
- The output record size is exceeded.
- OUTPUT directs the output record to be written (SKIP parameter).

The data written to the output record is written in its internal format. This circumstance can be altered by use of the EDIT function or by moving the data to a defined field.

Examples

To output bundle identifier and bundle status, use the following statements:

```
/OUTPUT BID  
/OUTPUT BSTAT
```

To output distribution identifier in column 1 and the first address line in column 35, specify the following OUTPUT control statements:

```
/OUTPUT DID COL(1)  
/OUTPUT A1 COL(35)
```

To output distribution identifier and all address lines as separate records, use:

```
/DEFINE I BIN  
/DO I = 1 TO NA BY 1  
/      OUTPUT DID SKIP  
/      OUTPUT A  
/      NEXT A  
/END
```

PRINT Control Statement

The PRINT control statement is used similarly to the OUTPUT control statement. The PRINT control statement writes data to the print file.

Syntax

```
PRINT [expression-1 [expression-2]] [COL(expression)] [SKIP(expression)] [PAGE]
```

where:

expression-1

Specifies an expression that determines the data to be placed in the print record.

expression-2

Specifies an expression that determines the data to be used as subheadings for expression-1.

If expression-2 contains a comma (for example, 'REPORT,IDENTIFIER'), the data is split at that point and written on separate heading lines. The heading data is aligned to the same position as expression-1.

COL(expression)

Specifies an expression that determines the position on the output record that expression-1 is written.

A value of 1 represents the first print position (character after the carriage control character). If this value is less than the current column position, the current print record is written to the print file, and expression-1 is placed in the new print record.

If COL is specified without an expression, the data is positioned directly after the previous print data.

If COL(expression) is omitted, expression-1 is positioned two characters from the previous print data.

SKIP(expression)

Specifies an expression that determines the number of lines to be skipped before printing the next print record.

This specification causes the current print record to be written to the print file and the column position to be reset to the first column. If SKIP is specified without an expression, one line is skipped.

PAGE

Specifies that the next print record is to be printed at the top of a new page.

This specification also causes the current print record to be written to the print file and the column position to be reset to the first column.

Note: If both PAGE and SKIP(*expression*) are specified, SKIP(*expression*) is ignored.

The data from each PRINT control statement is queued up contiguously in the print record. The print record is written under the following conditions:

- Statement processing has completed processing for a given database record, and the next PRINT control statement does not specify the COL operand.
- The COL operand is specified with a value less than the current column position.
- The print record size is exceeded.
- The PRINT control statement directs the print record to be written (SKIP or PAGE is specified).

Examples

To cause a control break (skip to the top of a new page) when the first character of the report identifier changes, enter the following:

```
/IF SUBSTR(RID,1,1) ^= SUBSTR(PREV(RID),1,1)
/      PRINT PAGE
/END
```

To print the first special instruction line in column 3 after skipping two lines, enter the following:

```
/PRINT INST1 COL(3) SKIP(2)
```

To print job name, step name, procedure step name, ddname, and report identifier with meaningful headings, enter the following statements:

```
/PRINT JOB 'JOB,NAME'
/PRINT STEP 'STEP,NAME'
/PRINT PROCSTEP 'PROCEDURE,STEP NAME'
/PRINT DD 'DD,NAME'
/PRINT RID 'REPORT,IDENTIFIER'
```

To print a descriptive title preceding the field being printed, specify:

```
/PRINT 'CLASS=' || CLASS
```

The same data can be printed using separate statements as follows:

```
/PRINT 'CLASS='
/PRINT CLASS COL
```

RELEASE Control Statement

The RELEASE control statement is used to signal the construction and release of a sort record to the sort. The actual data released to the sort is dependent on the current values of database fields and defined fields. The RELEASE control statement is only allowed in logic before the sorting (statements preceding the SORT control statement).

The RELEASE control statement is designed to reduce the amount of records that are released to the sort and to allow sorting of iterative fields, such as TEXT (report text line), INST (report instruction line), BINST (bundle instruction line), and A (distribution address line).

If the logic before the sorting drops through to the SORT control statement, an implied release is assumed. Be careful when coding the RELEASE control statement to avoid the release of identical information.

Syntax

```
RELEASE
```

Example

The following example restricts the sort selection to production jobs by interrogating the first four positions of the job description.

```
/DO FOREVER  
/  IF SUBSTR(JDESC,1,4) = 'PROD'  
/    RELEASE  
/  END  
  
/NEXT RECORD  
/END  
/SORT RID  
/PRINT RID  
/PRINT RDESC  
/END
```

SELECT Control Statement

The SELECT control statement is used to restrict statement processing to certain database records. The placement of the SELECT control statement is crucial, because it is interrogated at its relative position in the control statement flow. Typically, the SELECT control statement is placed at the beginning of the control statement flow or after the SORT control statement. Only one SELECT control statement is allowed, and it cannot be embedded within an IF or DO control statement.

Syntax

SELECT *condition*

where *condition* specifies the condition that is to be checked to determine which records to select from the database.

Example

To process only the job names that begin with P1, specify the following:

```
/SELECT SUBSTR(JOB,1,2) = 'P1'
```

Syntax

SELECT *jobname*

where *jobname* specifies the name of the job to be selected.

The SELECT function name can be abbreviated as S.

Example

The production JCL library containing all the production jobs is named PROD.JCLLIB. All jobs other than TEST1, TEST2, and TEST47 in the library are processed by the utility and job, and report description data for the jobs is added to the database. The cataloged procedure libraries that can be used by the production jobs are named SYS1.PROCLIB, PROD.PROCLIB, IPO1.PROCLIB, and TEST.PROCLIB. All SYSUDUMP and SYSABEND SYSOUT DD statements are omitted from the data added to the database.

The following job, which is located in RMOJCL in CAI.CVDEJCL, is executed:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//JCL      EXEC PGM=RMOJCL
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//JOBJCL   DD DSN=PROD.JCLLIB,DISP=SHR
//PROCLIB  DD DSN=SYS1.PROCLIB,DISP=SHR
//         DD DSN=PROD.PROCLIB,DISP=SHR
//         DD DSN=IPO1.PROCLIB,DISP=SHR
//         DD DSN=TEST.PROCLIB,DISP=SHR
//SYSIN    DD *
```

```
NAME DELIVER.SYSTEM1
EXCLUDE TEST1
EXCLUDE TEST2
EXCLUDE TEST47
OMIT SYSUDUMP
OMIT SYSABEND
//
```

Example

The production JCL library containing all of the production jobs is named PROD.JCLLIB. All jobs other than TEST1, TEST2, and TEST47 in the library are processed by the utility. Job and report description data for the jobs is added to the database. The procedure libraries that can be used by the production jobs are named SYS1.PROCLIB, PROD.PROCLIB, IP01.PROCLIB and TEST.PROCLIB. All SYSUDUMP and SYSABEND SYSOUT DD statements are omitted from the data added to the database.

The following job, which is member RMOJCS in CAI.CVDEJCL, is executed:

```
//EXAMPLE      JOB ACCOUNT,PROGRAMMER
//JCL   EXEC  PGM=RMOJCS
//STEPLIB      DD  DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT      DD  SYSOUT=A
//JOBJCLDD     DSN=PROD.JCLLIB,DISP=SHR
//PROCLIB      DD  DSN=SYS1.PROCLIB,DISP=SHR
//           DD  DSN=PROD.PROCLIB,DISP=SHR
//           DD  DSN=IP01.PROCLIB,DISP=SHR
//           DD  DSN=TEST.PROCLIB,DISP=SHR
//SYSIN DD     *

NAME  DELIVER.SYSTEM1
EXCLUDE TEST1
EXCLUDE TEST2
EXCLUDE TEST47
OMIT  SYSUDUMP
OMIT  SYSABEND
//
```

SET Control Statement

The SET control statement is used to set a define field to a specific value.

Syntax

```
SET field=expression-1
```

where:

expression-1

Specifies an expression that determines the data or value to be placed in the defined field.

Examples

To set field DEPT to positions 1–20 of address line 2, enter the following:

```
/DEFINE DEPT CHAR(20)  
/SET DEPT = A2
```

Similarly, if DEPT was in columns 21–40 of address line 2, you would extract this information as follows:

```
/DEFINE DEPT CHAR(20)  
/SET DEPT = SUBSTR(A2,21,20)
```

SORT Control Statement

The SORT control statement orders information that is eventually written to the print and output files. Capabilities are provided to sort up to 15 fields in ascending or descending sequence.

Syntax

```
SORT field[-seq][,field[-seq]], ...]
```

where:

field

Specifies the name of the field to be sorted.

This field can be a database field, a defined field, or a reserved field.

-seq

Specifies the sequence for sorting the field.

The sequence A is used for sorting in ascending sequence; the sequence D is used for sorting in descending sequence. If omitted, the field is sorted in ascending sequence.

The SORT control statement cannot be embedded within IF or DO control statements. Statements supplied before the SORT control statement are considered statements that are executed before the sort record is released to the sort, and statements following the SORT control statement are statements that are executed after the sort record is returned from the sort.

Sorting is not necessary for a majority of report requests. The database currently maintains crucial information in sequential order. The sequence of data retrieved from the database without sorting is determined by the following conditions:

- If database fields from the job descriptor record are referenced, database records are retrieved in job name (JOB) sequence.

- If database fields from the report descriptor record are referenced other than report identifier (RID), database records are retrieved in report identifier (RID) sequence.
- If database fields from the bundle descriptor record or bundle active status record are referenced without referencing database fields from the job descriptor record, report descriptor record, or report active status record other than report identifier (RID), database records are retrieved in bundle identifier (BID) sequence.
- If database fields are referenced from the distribution data record only, database records are retrieved in distribution identifier (DID) sequence.

Because the database fields that are referenced do not necessarily imply the true intent of your request, the sequence in which database records are accessed can be overridden by the CONTROL control statement. See the CONTROL control statement for further information.

Examples

To sort distribution identifiers and report identifiers in ascending sequence for a cross-reference report, enter the following:

```
/SORT DID, RID
```

The same sort criteria can be given by specifying the sort sequence:

```
/SORT DID-A, RID-A
```

To print a list of report identifiers with the amount of distribution identifiers from the largest to the smallest, specify the following:

```
/SORT NDID-D, RID-A  
/PRINT RID 'REPORT,IDENTIFIER'  
/PRINT NDID 'NUM,DISTIDS'
```

STOP Control Statement

The STOP control statement signals the end of a processing phase and the start of the next processing phase, if applicable. The processing phases (in order) are:

- Non-sort logic or logic before the sorting (statements preceding the SORT control statement)
- End-of-data logic (ON control statement) for non-sort or pre-sort logic
- Post-sort logic (statements following the SORT control statement)

- End-of-data logic (ON control statement) for post-sort logic

The pre-sort and post-sort designations are not applicable if sorting is not requested (no SORT control statement).

Syntax

STOP

Example

The following example concludes processing when the first digit of JOB is greater than or equal to "B":

```
/IF SUBSTR(JOB,1,1) GE 'B'  
/      STOP  
/END  
/PRINT JOB
```

THEN Control Statement

The THEN control statement is used with the IF control statement to specify the statements that are to be executed when a true condition is determined for the IF control statement. The THEN control statement does not have to be specified, but if it is, it can be specified at the end of the IF control statement specification or as a separate statement.

Syntax

THEN

If the THEN control statement is specified as a separated statement, this control statement must directly follow the corresponding IF control statement.

TITLE Control Statement

The TITLE control statement is used to define a report title for the printed report.

Syntax

TITLE *expression-1*

where *expression-1* specifies the expression whose data is used for the report title. This data is printed on the second line of the printed report and centered between the margins.

The TITLE control statement does not cause the report to eject to the top of a new page. The data for the TITLE control statement is saved and printed with each subsequent page. To force a page break, use the PRINT control statement (for example, PRINT PAGE). This PRINT control statement is specified before the TITLE control statement to ensure that the current report record is written. If the TITLE control statement is not supplied, a report title of 'GENERAL REPORT WRITER UTILITY' is used.

Examples

To set a report title of REPORT ATTRIBUTES, you enter:

```
TITLE 'REPORT ATTRIBUTES'
```

If you want to include the report identifier with a title similar to the previous example, you enter:

```
TITLE 'REPORT ATTRIBUTES FOR' ||RID
```


Example 1

The following statements, which are located in RMOGRW01 in CAI.CVDEOPT, produce a report that contains a job name, step name, procedure step name, ddname, report identifier, report type, number of text entries for report, number of dynamic variables and active status. The report is produced with headings and a title. Each job name is printed on a new page.

```
//EXAMPLE1 JOB ACCOUNT,PROGRAMMER
//*
//*  RMOGRW - GENERAL PURPOSE REPORT UTILITY
//*
//RMOGRW  EXEC PGM=RMOGRW
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR          <=== MODIFY
//SYSPRINT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//PRTFILE DD SYSOUT=*
//*
/* SAMPLE CONTROL STATEMENTS TO GENERATE A REPORT CONTAINING
/*      JOB NAME, STEP NAME, PROCEDURE STEP NAME, DD NAME, REPORT
/*      IDENTIFIER, REPORT TYPE AND ACTIVE STATUS. EACH JOB IS ON
/*      A NEW PAGE AND TITLE AND HEADINGS ARE PRODUCED.
/*
//SYSIN   DD *
/CONTROL DATABASE=DELIVER.SYSTEM1
/PRINT PAGE
/TITLE 'REPORTS FOR JOB '||JOB
/DEFINE I BIN
/DO I = 1 TO NRID BY 1
/  PRINT STEP 'STEP,NAME' SKIP(1)
/  PRINT PROCSTEP 'PROCEDURE,  NAME'
/  PRINT DD ' ,DDNAME'
/  PRINT RID 'REPORT,IDENTIFIER'
/  PRINT TRANS(TYPE,'M','MONITR',
               'S','STACK ',
               'I','INTER ',
               'C','CNTL ',
               'D','DYNAM ',
               * , '      ') 'REPORT, TYPE'
/  PRINT NTEXT 'TXT,NO#'
/  PRINT NVAR 'DYN,VAR#'
/  PRINT TRANS(ACTIVE,'Y','YES',
               * , '      ') 'ACTIVE'
/  NEXT RID
/END
/*
//
```

Example 1 Output

The following panel illustrates the report produced by Example 1:

02/28/2013 16:41:02 RMOGRW		CA DELIVER REPORTS FOR JOB APDAILY1		PAGE 1	
STEP NAME	PROCEDURE NAME	DDNAME	REPORT IDENTIFIER	REPORT TYPE	ACTIVE
STEP1		REPORT	ACCOUNTS PAYABLE DAILY REPORT 01		YES
STEP2		REPORT	ACCOUNTS PAYABLE DAILY REPORT 02		YES
STEP3		REPORT	ACCOUNTS PAYABLE DAILY REPORT 03		
STEP4		REPORT	ACCOUNTS PAYABLE DAILY REPORT 04	STACK	
		REPORT	ACCOUNTS PAYABLE DAILY REPORT 05	STACK	
		REPORT	ACCOUNTS PAYABLE DAILY REPORT 06	STACK	
STEP5		REPORT	ACCOUNTS PAYABLE DAILY REPORT 07		

Example 2

A formatted report containing report identifiers, report attributes, special instructions, text specifications, and distribution identifiers is produced from this example, which is located in RMOGRW02 in CAI.CVDEOPT.

Each report identifier appears on a separate page.

```
//EXAMPLE2 JOB ACCOUNT,PROGRAMMER
//*
//*   RMOGRW - GENERAL PURPOSE REPORT UTILITY
//*
//RMOGRW   EXEC PGM=RMOGRW
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR          <=== MODIFY
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//PRTFILE  DD SYSOUT=*
//*
//* SAMPLE CONTROL STATEMENTS TO GENERATE A REPORT CONTAINING
//*       REPORT IDENTIFIERS, REPORT ATTRIBUTES, SPECIAL INSTRUCTIONS,
//*       TEXT SPECIFICATIONS, AND DISTRIBUTION IDENTIFIERS. EACH
//*       REPORT IDENTIFIER APPEARS ON A SEPARATE PAGE.
//*
//SYSIN    DD *
/CONTROL DATABASE=DELIVER.SYSTEM1
/PRINT PAGE
/TITLE 'REPORT DEFINITION ATTRIBUTES'
/PRINT 'ID='||RID COL(1)
/PRINT 'JOB='||JOB COL(37)
/PRINT TRANS(TYPE,      'M','TYPE=MONITR',
                  'S','TYPE=STACK ',
                  'I','TYPE=INTER ',
                  'C','TYPE=CNTRL ',
                  'D','TYPE=DYNAM ',
                  * , 'TYPE=      ') COL(51)
/PRINT 'STEP='||STEP COL(1)
/PRINT 'PROCSTEP='||PROCSTEP COL(26)
/PRINT 'DDNAME='||DD COL(51)
/PRINT 'REPORT ATTRIBUTES:' SKIP(2)
/PRINT 'ARCH='||ARCH COL(3)
/PRINT 'BANNER='||BANNER COL(26)
/PRINT 'BURST='||BURST COL(51)
/PRINT 'CC='||CC COL(3)
/PRINT 'CLASS='||CLASS COL(26)
/PRINT 'COPIES='||COPIES COL(51)
/PRINT 'FCB='||FCB COL(3)
/PRINT 'FORM='||FORM COL(26)
/PRINT 'FLASH=('||FLASH||','||
              EDIT(FLASHCT,'ZZZ')||')' COL(51) /PRINT 'LATE='||LATE COL(3)
/PRINT 'OPTCDJ='||OPTCDJ COL(26)
/PRINT 'UCS='||UCS COL(51)
/PRINT 'CHARS=('||      CHARS1||','||
                      CHARS2||','||
                      CHARS3||','||
                      CHARS4||')' COL(3)
/PRINT 'MODIFY=('||MODIFY||','||
```

```
        EDIT(TRC, 'ZZZ') || ') ' COL(51)
/PRINT 'COPYG=( ' ||
        EDIT(COPYG1, 'ZZZ') || ', ' ||
        EDIT(COPYG2, 'ZZZ') || ', ' ||
        EDIT(COPYG3, 'ZZZ') || ', ' ||
        EDIT(COPYG4, 'ZZZ') || ', ' ||
        EDIT(COPYG5, 'ZZZ') || ', ' ||
        EDIT(COPYG6, 'ZZZ') || ', ' ||
        EDIT(COPYG7, 'ZZZ') || ', ' ||
        EDIT(COPYG8, 'ZZZ') || ') ' COL(3)
/DEFINE I BIN
/PRINT 'TEXT SPECIFICATIONS:' SKIP(2)
/DO I = 1 TO NTEXT BY 1
/  PRINT TEXTTYPE          COL(3)
/  PRINT TEXTREUSE         COL(8)
/  PRINT TEXTOP            COL(11)
/  PRINT EDIT(TEXTLINE, 'ZZZ') COL(15)
/  PRINT EDIT(TEXTCOL, 'ZZZ') COL(20)
/  PRINT TEXT              COL(25)
/  NEXT TEXT
/END
/PRINT 'Dynamic Variables:' SKIP(2)
/DO I = 1 TO NVAR BY 1
/  PRINT VARNAME           COL(3)
/  PRINT EDIT(VARLEN, 'ZZZ') COL(12)
/  PRINT EDIT(VARLINE, 'ZZZ') COL(16)
/  PRINT EDIT(VARCOL, 'ZZZ') COL(20)
/  NEXT VARNAME
/END
/PRINT 'SPECIAL INSTRUCTIONS:' SKIP(2)
/DO I = 1 TO NINST BY 1
/  PRINT INST COL(3)
/  NEXT INST
/END
/PRINT 'DISTRIBUTION IDENTIFIERS:' SKIP(2) /DO I = 1 TO NDID BY 1
/  PRINT GROUP COL(3)
/  PRINT DISTID
/  PRINT DCOPIES
/  PRINT DEST
/  NEXT DISTID
/  END
/*
//
```

Example 2 Output

The following panel illustrates the report produced by Example 2:

02/28/2013 16:42:32	CA DELIVER	PAGE 1
RMOGRW	REPORT DEFINITION ATTRIBUTES	
ID=ACCOUNTS PAYABLE DAILY REPORT 01	JOB=APDAILY1	TYPE=
STEP=STEP1	PROCSTEP=	DDNAME=REPORT
REPORT ATTRIBUTES:		
ARCH=	BANNER=	BURST=N
CC=	CLASS=	COPIES=
FCB=	FORM=	FLASH=(,)
LATE=	OPTCDJ=	UCS=
CHARS=(, , , ,)		MODIFY(,)
COPYG=(, , , , , , , ,)		
TEXT SPECIFICATIONS:		
SPECIAL INSTRUCTIONS:		
DISTRIBUTION IDENTIFIERS:		
ACCOUNTS PAYABLE DEPARTMENT DROP		

Example 3

The following example, which is located in RMOGRW03 in CAI.CVDEJCL, lists the address lines and destination for each distribution identifier:

```
//EXAMPLE3 JOB ACCOUNT,PROGRAMMER
//RMOGRW3 EXEC PGM=RMOGRW
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//PRTFILE DD SYSOUT=*
//SYSIN DD *

/CONTROL DATABASE=DELIVER.SYSTEM1
/TITLE 'DISTRIBUTION SPECIFICATIONS'
/PRINT DISTID 'DISTRIBUTION, IDENTIFIER' SKIP(2)
/PRINT DDEST 'DISTRIBUTION,DESTINATION'
/DEFINE (DCOL,I) BIN
/SET DCOL = COL+2
/DO I = 1 TO NA BY 1
/ PRINT A 'DISTRIBUTION, LINES' COL(DCOL)
/ NEXT A
/END
```

Example 3 Output

The following panel illustrates the report produced by Example 3:

02/28/2013 17:24:33 RMOGRW	CA DELIVER DISTRIBUTION SPECIFICATIONS	PAGE 7
DISTRIBUTION IDENTIFIER	DISTRIBUTION DESTINATION	DISTRIBUTION LINES
ACCOUNTS PAYABLE DEPARTMENT DROP	LOCAL	DELIVER TO ACCOUNTS PAYABLE DEPT
ACCOUNTS RECEIVABLE DEPARTMENT	LOCAL	DELIVER TO ACCOUNTS RECEIVABLE

Example 4

The following example, which is located in RMOGRW04 in CAI.CVDEJCL, produces a report containing bundle identifiers, bundle attributes, special instructions, report identifiers, and distribution identifiers. Each bundle identifier appears on a separate page. Report identifiers and distribution identifiers print horizontally.

```
//EXAMPLE4 JOB ACCOUNT,PROGRAMMER
//RMOGRW EXEC PGM=RMOGRW
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//PRTFILE DD SYSOUT=*
//SYSIN DD *

/CONTROL DATABASE=DELIVER.SYSTEM1
/PRINT PAGE
/TITLE 'BUNDLE DEFINITION FOR '|||BID
/PRINT 'ATTRIBUTES:' SKIP(2)
/PRINT 'BANNER=('|||BBANNER1|||', '|||
        BBANNER2|||', '|||
        BBANNER3|||)' COL(3)
/PRINT 'DIST='|||BDIST COL(3)
/DEFINE I BIN
/DO I = 1 TO NBA BY 1
/ PRINT BA COL(6)
/ NEXT BA
/END

/PRINT 'LATE='|||BLATE COL(3)
/PRINT 'INTERVAL='|||BINVL COL(3)
/PRINT 'SPECIAL INSTRUCTIONS:' SKIP(2)
/DO I = 1 TO NBINST BY 1
/ PRINT BINST COL(3)
/ NEXT BINST
/END
```

```

/DEFINE PCOL BIN
/SET PCOL = 999
/PRINT 'REPORT IDENTIFIERS:' SKIP(2)
/DO I = 1 TO NRID BY 1
/  IF PCOL+32 > LINESIZE-1
/    SET PCOL = 3
/  END
/  PRINT RID COL(PCOL)
/  SET PCOL=PCOL+34
/  NEXT RID
/END

/SET PCOL = 999
/PRINT 'DISTRIBUTION IDENTIFIERS:' SKIP(2)
/DO I = 1 TO NDISTID BY 1
/  IF PCOL+32 > LINESIZE-1
/    SET PCOL = 3
/  END
/  PRINT DISTID COL(PCOL)
/  SET PCOL=PCOL+34
/  NEXT DISTID
/END

/PRINT 'JOB CARDS:' SKIP(2)
/PRINT BJCD1 COL(3)
/PRINT BJCD2 COL(3)
/PRINT BJCD3 COL(3)
/PRINT BJCD4 COL(3)
/END

```

Example 4 Output

The following panel illustrates the report produced by Example 4:

```

02/28/2013 18:06:25          CA DELIVER          PAGE  2
RMOGRW          BUNDLE_DEFINITION_FOR_ACCOUNTS_PAYABLE_DAILY_BUNDLE_01

ATTRIBUTES:
  BANNER=(          ,          ,          )
  DIST=ACCOUNTS_PAYABLE_DEPARTMENT_DROP
  LATE=

SPECIAL INSTRUCTIONS:

REPORT IDENTIFIERS:
  ACCOUNTS_PAYABLE_DAILY_REPORT_01  ACCOUNTS_PAYABLE_DAILY_REPORT_02  ...
  ACCOUNTS_PAYABLE_DAILY_REPORT_04  ACCOUNTS_PAYABLE_DAILY_REPORT_07

DISTRIBUTION IDENTIFIERS:
  ACCOUNTS_PAYABLE_DEPARTMENT_DROP

JOB CARDS:
  //APBNL02 JOB ACCOUNTN,PROGRAMMER

```

Example 5

The following example, which is located in RMOGRW05 in CAI.CVDEJCL, produces an Archive ID (ARCHID) / Report ID Cross-Reference report. It lists each ARCHID and the Deliver reports that use this ID.

```
//EXAMPLE5 JOB ACCOUNT,PROGRAMMER
//*
//*  RMOGRW - GENERAL PURPOSE REPORT UTILITY
//*
//RMOGRW EXEC PGM=RMOGRW
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR      <=== MODIFY
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//PRTFILE DD SYSOUT=*
//*
//SYSIN DD *
/CONTROL DATABASE=DELIVER.SYSTEM1
* LISTING OF ALL REPORTS FOR AN ARCHID
/SELECT ARCHID NE ' '
/TITLE 'ARCHID / REPORT ID CROSS-REFERENCE'
/SORT ARCHID
* START POST SORT LOGIC
/DEFINE PCOL BIN(2)
/DEFINE PARCHID CHAR(32)
/DO FOREVER
/  IF ARCHID NE PARCHID
/    SET PARCHID = ARCHID
/    PRINT ARCHID,'ARCHIVE-ID' COL(1)
/    SET PCOL = 33
/  END
/  IF PCOL+32 > LINESIZE
/    PRINT ' ' COL(32)
/    SET PCOL = 33
/  END
/  PRINT RID,'REPORT-ID'
/  SET PCOL = PCOL + 33
/  NEXT RECORD
/END
/ON ENDDATA
/ PRINT SKIP
/END
```


Example 5 Output

The following panel illustrates the report produced by Example 5:

10/14/2013 14:10:01	CA Deliver (12.1)		
RMOGRW	ARCHID / REPORT ID CROSS-REFERENCE		
ARCHIVE-ID	REPORT-ID	REPORT-ID	REPORT-ID
AR STATEMENTS	ARWEEKLY-R01	ARMONTHLY-R01	ARYEARLY-R01
PROFIT LOSS REPORT	GLSALES-R01	INVSUM-R02	

RMOHTP - Batch Detail History Reporting

The RMOHTP utility can be used for detail history tracking. RMOHTP tracks bundles and reports to a maximum of five stations using default data or data specified in control cards.

Job Control Statements

Specify the following JCL to execute RMOHTP:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMOHTP) and, optionally, the high-level name of the database as the PARM parameter (PARM='DELIVER. SYSTEM1').

STEPLIB DD

.Identifies the load library that contains RMOHTP.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

STATIONn DD

Defines the 1–5 sequential data sets (where n=1–5) containing tracking data for posting reports and bundles at station n.

Parameters specified in this statement override default data.

DATA Control Statement

The DATA control statement is used to specify the default user data for posting to the detailed data. If you do not specify a DATA control statement, the default user data for posting is the job name.

Syntax

```
DATA xxxxxxxxxxxx
```

where xxxxxxxxxxxx specifies a maximum of 12 characters of the user data.

DATE Control Statement

The DATE control statement is used to specify the default date for posting to the detailed historical data. If you do not specify a DATE control statement, the default date for the posting is the system date.

Syntax

```
DATE xxxxxxxxxx
```

where xxxxxxxxxx specifies the posting default date set in RMODFMT (for example, 02/20/2002).

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database. If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used.

TIME Control Statement

The TIME control statement is used to specify the default time for posting to the detailed historical data. If you do not specify a TIME control statement, the default time for the posting is the system time.

Syntax

```
TIME xxxxxxxx
```

where xxxxxxxx specifies the posting default time in a *hh:mm:ss* format (for example, 08:30:00).

Station Data

Station data defining the reports and bundles by station that are to be posted are input from sequential data sets defined by DD statements STATION*n* (for example, data for station 1 would be input from DD statement STATION1, STATION2, ...).

The records that read input contain the following fields:

Column	Contents
1–10	History detail number (HDN)
11	blank
12–21	Posting date in the RMODFMT default date format (for example, 10/17/2013)
22	blank
23–30	Posting time in a <i>hh:mm:ss</i> format (for example, 10:30:24)
31	blank
32–43	User data

The history detail number is the only field of the record that is required. The default values as defined by the other control statements such as data and time are used for any field omitted from the record.

Example

To initiate the RMOHTP batch detail history tracking program, you execute the following job, which is located in RMOHTP in CAI.CVDEJCL:

```
//EXAMPLE1 JOB ACCOUNT,PROGRAMMER
//TRACK EXEC PGM=RMOHTP
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=BLKSIZE=133
//SYSIN DD *

NAME DELIVER.SYSTEM1
DATA BATCH
//STATION1 DD *
0000230100
0000230200 10/23/2013
//STATION2 DD *
0000230100 10/23/2013
0000230200 10/24/2013
//
```

RMOIFMAP - Index File Mapping Utility

The RMOIFMAP diagnostic utility is used by CA Technical Support to assess reports of index errors in a database. The output lists index levels, keys, and error conditions.

Sample JCL

To run RMOIFMAP, specify the following JCL:

```
//... JOB...  
//STEP1 EXEC PGM=RMOIFMAP,PARM='DELIVER.SYSTEM1'  
//STEPLIB DD DISP=SHR,DSN=CAI.CVDELOAD  
//SYSPRINT DD SYSOUT=*
```

where:

DELIVER.SYSTEM1 is the high-level name of your database

CAI.CVDELOAD is the name of the load library that contains the RMOIFMAP program

If this library is link-listed, you can omit the STEPLIB DD statement.

The data written to SYSPRINT is the information that CA Technical Support needs to resolve your issue.

RMOJCL - Automatic Database Construction From JCL

RMOJCL automatically constructs the database by scanning your job JCL library and adding job and report description data to the database. This utility allows you to quickly and easily construct your database so that you can immediately begin using the CA Deliver system.

The utility reads job JCL, expands the cataloged and in stream procedures, performs symbolic substitution, and scans the JCL for SYSOUT DD statements. It then adds data to the database for each SYSOUT data set and produces a report detailing the SYSOUT data sets. Also, control statements are available to restrict the jobs processed and to omit certain types of SYSOUT DD statements.

Note: The maximum number of reports that the RMOJCL utility can process per job is 9999.

The data that is added to the database consists of the following:

- Job name
- Step name
- Procedure step name
- DDname
- Report identifier
- Report type

Syntax

`jobname-Rnn`

where:

jobname

Specifies the name of the job containing the SYSOUT data set.

Rnn

Specifies a numeric sequence number within the job starting with 01 and incremented by 1.

Note: When *nn* exceeds 99, *R* is omitted from the report identifier; when *n* exceeds 999, the dash (-) is omitted.

For example, the first SYSOUT DD statement for job S27P35 would be assigned a report identifier of S27P35-R01, the second DD statement for the job would be assigned a report identifier of S27P35-R02, and so on.

The report type that is assigned is monitored data output.

User exit RMOJCLUX is available to review and/or modify the data before its being added to the database. For example, if you want a different format for the report identifiers, you could apply a different format through the user exit. For more information, see the chapter "User Exits" in the *Programming Guide*.

Note: The job JCL and associated cataloged procedures are assumed free of errors. As such, no indication is given when an error condition is encountered. For example, if the utility is unable to find a cataloged procedure, the utility continues with the next input JCL statement as if no error had occurred.

Job Control Statements

Specify the following JCL to execute RMOJCL:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMOJCL) and, optionally, the high-level name of the database as the PARM parameter (PARM='DELIVER.SYSTEM1').

STEPLIB DD

Identifies the load library that contains RMOJCL.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

JOBCL DD

Defines the sequential or partitioned data set containing the JCL for the jobs to be input.

For a sequential data set, the jobs is stacked together; for a partitioned data set, each member contains one job with the member name equal to the name of the job. In both cases, the actual job name used is the one contained on the JCL JOB statement.

PROCLIB DD

Defines the cataloged procedure libraries.

Multiple libraries can be specified by concatenating the DD statements for the libraries. Be sure to specify all the libraries that can be required. The procedure libraries in use at your installation can be determined by examining the JCL procedure in SYS1.PROCLIB for starting JES2 or JES3.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

- The keyword TEST can be coded as the second subparameter of the PARM field on the EXEC statement to indicate that a test run of the utility is to be made (for example, PARM='DELIVER.SYSTEM1,TEST'). A test run scans the JCL and produces a detailed report of the SYSOUT DD statements. No data, however, is added to the database.

- The keyword CHECK can be coded as the second subparameter of the PARM field on the EXEC statement (for example, PARM='DELIVER.SYSTEM1,CHECK') to compare job definitions in the database to the production JCL.

Note: An RMOJCL control statement must fit entirely on one card image. Continuations are not allowed.

EXCLUDE Control Statement

The EXCLUDE control statement is used to exclude a job from processing. Only one job can be specified per EXCLUDE control statement; however, as many statements as necessary can be used.

Syntax

```
EXCLUDE jobname
```

where *jobname* specifies the name of the job to be excluded.

The EXCLUDE function name can be abbreviated as E.

Syntax

```
EXCLUDE jobname
```

where *jobname* specifies the name of the job to be excluded.

The EXCLUDE function name can be abbreviated as E.

Important! An RMOJCS control statement must fit entirely on one card image—continuations are not allowed.

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database. If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used.

OMIT Control Statement

The OMIT control statement is used to omit specified types of SYSOUT DD statements from processing. As many statements as necessary can be used.

Syntax

`OMIT ddname pgmname class`

where:

ddname

Specifies the name of the DD statement to which the control statement applies.

You can specify an asterisk to indicate that the control statement applies to all DD statements.

pgmname

Specifies the program name, as in the PGM= parameter on the EXEC JCL statements, to which the OMIT control statement applies.

If omitted, the control statement applies to all programs.

You can specify an asterisk to indicate that the control statement applies to all program names.

class

Specifies the SYSOUT class to which the OMIT control statement applies.

If omitted, the control statement applies to all SYSOUT classes.

The OMIT function name can be abbreviated as O.

Examples

To omit adding report definitions for any SYSPRINT DD statement written to by the IDCAMS utility, enter the following:

```
OMIT SYSPRINT IDCAMS
```

To omit adding report definitions for any SYSUDUMP DD statement, enter the following:

```
OMIT SYSUDUMP
```

Syntax

`OMIT ddname class`

where:

ddname

Specifies the name of the DD statement to which the control statement applies. You can specify an asterisk to indicate that the control statement applies to all DD statements.

class

Specifies the SYSOUT class to which the OMIT control statement applies. If omitted, the control statement applies to all SYSOUT classes.

You can abbreviate the OMIT function name as O.

Examples

To omit adding report definitions for any SYSPRINT DD statement written to class X, enter the following:

```
OMIT SYSPRINT X
```

To omit adding report definitions for any SYSUDUMP DD statement, enter the following:

```
OMIT SYSUDUMP
```

SELECT Control Statement

The SELECT control statement is used to select a specific job for processing; any other job input to the utility that is not also selected for processing is excluded from processing. You can specify only one job per SELECT control statement; however, you can use as many statements as necessary.

RMOJCS - Enhanced Database Construction from JCL

The RMOJCS utility automatically constructs the database by scanning your job JCL library and adding job and report description data to the database.

This utility allows you to construct your database quickly and easily so that you can immediately begin using the CA Deliver system. It is an enhancement of the original RMOJCL utility in that it uses the common component of CA JCLCheck Utility (CA JCLCheck) to scan your PROC and JCL libraries so that CA JCLCheck is compatible with all of the latest changes in the Job Control Language.

Note: The JCLCheck Common Component is no longer distributed as part of the CA Deliver Distribution Libraries. For more information about the RMOJCS Utility, see the CA JCLCheck Common Component Installation documentation.

If both of the following conditions are true, continue to use the original RMOJCL utility:

- You are not using any of the new features of Job Control Language

- You must omit SYSOUTs based on the program, or you want to exclude SYSOUTs based on the program or PROC name in the RMOJCLUX user exit. (RMOJCS is unable to provide program and PROC names.)

The RMOJCS utility reads the JCL and calls the JCLCheck common component to extract a list of the jobs, steps, procedure steps, and SYSOUT DDs in the input data. It then adds the following data to the database for each SYSOUT data set and produces a report detailing the SYSOUT data sets:

- Job name
- Step name
- Procedure step name
- DDname
- Report identifier
- Report type

Also, control statements are available to restrict the jobs processed and to omit certain types of SYSOUT DD statements. The maximum number of reports that the RMOJCS utility can process per job is 9999.

Syntax

Jobname-Rnn

where:

jobname

Specifies the name of the job that contains the SYSOUT data set.

Rnn

Specifies the numeric sequence number within the job starting with 1 and incremented by 1.

For example, the first SYSOUT DD statement for job S27P35 would be assigned a report identifier of S27P35-R01, the second DD statement for the job would be assigned a report identifier of S27P35-R02, and so on.

Note: When nn exceeds 99, R is omitted from the report identifier; when nnn exceeds 999, the dash (-) is omitted.

The report type that is assigned is monitored data output.

The RMOJCLUX user exit is used to review, to modify, or both, the data before its being added to the database. For example, if you want a different format for the report identifiers, you could change the format through the RMOJCLUX user exit. For more information, see the chapter "User Exits" in the *Programming Guide*.

Note: The job JCL and associated cataloged procedures are assumed free of errors. As such, no indication is given when an error condition is encountered. For example, if the utility is unable to find a cataloged procedure, the utility continues with the next input JCL statement as though no error had occurred.

Job Control Statements

Specify the following JCL to execute RMOJCS:

JOB

Initiates the job.

EXEC

Specifies the name of the program (PGM=RMOJCS) and, optionally, the high-level name of the database as the PARM parameter.(PARM='DELIVER.SYSTEM1').

To indicate that a test run of the utility is to be made, you can code the keyword TEST as the second subparameter of the PARM field on the EXEC statement, for example, PARM='DELIVER.SYSTEM1,TEST'. A test run scans the JCL and produces a detailed report of the SYSOUT DD statements. However, no data is added to the database.

To compare job definitions in the database to the production JCL, you can code the keyword CHECK as the second subparameter of the PARM field on the EXEC statement, for example, PARM='DELIVER.SYSTEM1,CHECK'. When reports on jobs are found in the JCL but not in the database, and the reverse, an error is flagged.

To specify that the input JCL is to come from an

CA Librarian (AllFusion CA Librarian) library, code the keyword LIB as the third subparameter of the PARM field on the EXEC statement. For example, PARM='DELIVER.SYSTEM1,TEST,LIB' or PARM=','LIB' tells RMOJCS that the data set specified in the JOBJCL DD statement is a CA Librarian file.

To specify that the input JCL is to come from a

CA Panvalet (AllFusion CA-Panvalet) library, code the keyword PAN as the third subparameter of the PARM field on the EXEC statement. For example, PARM='DELIVER.SYSTEM1,TEST,PAN' or PARM=','PAN' tells RMOJCS that the data set specified in the JOBJCL DD statement is a CA Panvalet file.

STEPLIB DD

Identifies the load library that contains RMOJCS.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

JOBJCL DD

Defines the sequential or partitioned data set, the CA Librarian library, or the CA Panvalet library that contains the JCL for the jobs to be input to RMOJCS.

For a sequential data set, the jobs are stacked together. For a partitioned data set, a CA Librarian library, or a CA Panvalet library, each member contains one job with the member name equal to the name of the job. In both cases, the actual job name used is the one contained on the JCL JOB statement. Do not use the SUBSYS parameter for CA Librarian or CA Panvalet libraries.

If the JOBJCL data set contains members that are JOBS and members that are procedures, then use SELECT statements to insure that you process the JOBS that you want. If you do not use SELECT statements, then CA JCLCheck adds a default JOBCARD to each member that is a procedure that is in the JOBJCL library and then process the procedure as if it were a job with the same name as the procedure. This causes phantom JOB and REPORT definitions to be built in your CA Deliver database.

PROCLIB DD

Defines the cataloged procedure libraries.

Multiple libraries can be specified by concatenating the DD statements for the libraries. Be sure to specify all of the libraries that can be required. The procedure libraries in use at your installation can be determined by examining the JCL procedure in SYS1.PROCLIB for starting JES2 or JES3.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

EXCLUDE Control Statement

The EXCLUDE control statement is used to exclude a job from processing. Only one job can be specified per EXCLUDE control statement, however, as many statements as necessary can be used.

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database. If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used.

OMIT Control Statement

The OMIT control statement is used to omit specified types of SYSOUT DD statements from processing. As many statements as necessary can be used.

SELECT Control Statement

The SELECT control statement is used to select a specific job for processing; any other job input to the utility (but not also selected for processing) is excluded from processing. You can specify only one job per SELECT control statement, however, you can use as many statements as necessary.

RMOPRE - Take Action on the Most Recently Produced Reports

Use the RMOPRE utility if you have CA 11 and want to use it with CA Deliver.

The RMOPRE batch utility program allows CA Deliver to take action on the most recently produced reports that were generated by specified steps **before** an actual rerun job is executed. Use this batch utility program to process the most recently produced reports.

Reports on which you can act are:

- Bundle holding copies in the JES spool
- Bundle holding copies placed in the CA View database by a direct-to-SAR operation
- Non-bundled reports placed in the CA View database by a direct-to-SAR operation

Note: You cannot act on reports that the CA View archival task has archived in the CA View database.

Execute RMOPRE on the same system where the CA Deliver database you specify as a parameter to this utility program is located.

Important! Use RMORMS in the most recently produced job to use RMOPRE to back out reports. CA Deliver only backs out reports from the database specified in the SAR= initialization parameter.

Syntax

```
//prerun      EXEC PGM=RMOPRE,PARM='high-level-name'  
//SYSIN       DD      *  
NAME          high-level-name  
JOBNAME       job-name  
JOBNUM        job-number  
FROM-STEP     jobstep.procedurestep  
TO-STEP       jobstep.procedurestep  
EXCLUDE       jobstep.procedurestep  
LSERV-INBSSN  value
```

Using AFP ACIF to Archive Reports

The CA View ACIF interface, which allows you to archive AFP reports, does not accept direct-to-View reports. You cannot, therefore, automatically back out direct-to-View AFP ACIF reports using the standard interface between CA Deliver and CA View.

Setting up CA 11 (at a system level) to run with CA Deliver is described in the *Installation Guide*. Options you can specify online (for example, delete, flag, and keep) are described in the *Administration Guide*.

Job Control Statements

Specify the following JCL to execute RMOPRE:

EXEC

Specifies the name of the program (PGM=RMOPRE) and the PARM parameter.

The PARM parameter specifies the 1- to 17-character, high-level name of the database where the parameters are specified.

SYSIN DD

Specifies the name of the data set where the rerun control statements you want to input are located.

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database. CA Deliver searches databases for reports in the order you specify the databases and uses the first database that contains the executing job definition.

You can specify NAME more than once.

If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used.

JOBNAME Control Statement

The JOBNAME control statement is used to specify the name of the job that produces the report or reports you rerun.

Syntax

`JOBNAME jobname`

where *jobname* specifies the name of the job that produced the report or reports you rerun.

JOBNUM Control Statement

The JOBNUM control statement is used to specify the number, assigned by JES to a job as it enters the system that distinguishes one job from another.

Syntax

`JOBNUM job-number`

where *job-number* specifies the JES job number of the job that produced the report or reports you rerun.

FROM-STEP Control Statement

The FROM-STEP control statement is used to specify the job step and procedure step from which you want to rerun a job. The FROM-STEP you specify must be a step defined in the CA Deliver job definition.

This control statement is optional. If omitted, the first step in the job definition is used.

Syntax

`FROM-STEP jobstep,procstep`

where *jobstep,procstep* specifies the job step and the procedure step from which you want to rerun the job specified by the preceding JOB control statement. Specifying the procedure step is optional.

TO-STEP Control Statement

The TO-STEP control statement is used to specify the job step and procedure step to which you want to rerun a job. The TO-STEP you specify must be a step defined in the CA Deliver job definition.

This control statement is optional. If omitted, the last step in the job definition is used.

Syntax

TO-STEP *jobstep,procedurestep*

where *jobstep,procedurestep* specifies the job step and procedure step to which you want to rerun the job specified by the preceding JOB control statement. Specifying the procedure step is optional.

EXCLUDE Control Statement

The EXCLUDE control statement is used to exclude a step from processing.

You can specify EXCLUDE more than once.

LSERV-INBSSN Control Statement

The LSERV-INBSSN control statement is used to specify that CA 11 calls the CA Balancing rerun interface routine. If omitted, this utility does not back out CA Balancing data for corresponding CA Deliver reports.

Syntax

LSERV-INBSSN *value*

where *value* specifies a four-digit or four-character (alphanumeric) CA L-Serv Database Manager identifier, which identifies the subsystem name of the CA L-Serv that manages the CA Balancing database.

Note: For more information, see the *CA Balancing System Guide*, and the CA Common Services documentation.

Example

The following example illustrates how to use RMOPRE:

```
//PRERERUN    EXEC PGM=RMOPRE,PARM='DELIVER.SYSTEM1'
//RMOJTAB     DD   DSN=DELIVER.RMOJTAB,DISP=SHR
//SYSIN       DD   *
NAME          DELIVER.SYSTEM1
JOBNAME       JOB123
JOBNUM        1097
FROM-STEP     STEP1.PROC1
TO-STEP       STEP9
EXCLUDE       STEP3.PROC1
LSERV-INBSSN  INB1
```

Syntax

LSERV-INBSSN *value*

where *value* specifies a four-digit or four-character (alphanumeric) CA L-Serv identifier, which identifies the subsystem name of the CA L-Serv that manages the CA Balancing database.

CA Balancing is described in the *CA Balancing System Guide*. CA L-Serv is described in the CA Common Services documentation.

Example

The following shows how to use RMORMS:

```
//step1 EXEC PGM=RMORMS,PARM='DELIVER.SYSTEM1'  
//RMONET1 DD SYSOUT=x,DEST=dest,FREE=CLOSE  
//RMSPARMS DD DSN=DELIVER.RMSPARMS,DISP=SHR  
//SYSPRINT DD SYSOUT= *
```

Using Page and Form Definitions

This section tells you what happens to, and how to accommodate reports defined by page and form definition resources when those reports are placed under the control of CA Deliver.

You define page (PAGEDEF) and form (FORMDEF) definition resources that are suited to the particular SYSOUT data produced by your application program. When you place this SYSOUT data under the control of CA Deliver, CA Deliver inserts banner pages before and after the SYSOUT data.

AFP software is unable to distinguish between a banner page and original SYSOUT; therefore, AFP software processes banner pages according to the page and form definition resources you defined and intended for the original SYSOUT data. As a result, the banner pages are output in an unexpected format.

Note: Page and form definition resources are stored in data sets in libraries such as SYS1.PDEFLIB and SYS1.FDEFLIB, which are allocated to the AFP PSF started task.

Page and Form Definition Options

If your page and form definitions are affecting the SYSOUT data in your reports, you can do the following actions:

- Suppress the output of banner pages by entering an asterisk in the BANNER field on the Report Definition Attributes panel.

- Define your own in-stream AFP (Authorized Program Facility) commands embedded in X'5A' (Hex Five Able) records in the banner pages.
- Use RMOPSF, a stand-alone utility to modify members in which PAGEDEF and FORMDEF resources are defined or to create new members in which changes that accommodate banner pages are defined.

The last choice is the more viable because, in most cases, you must print banner pages with your report and defining your own in-stream AFP commands in X'5A' records requires extra effort.

How RMOPSF Works

RMOPSF accommodates CA Deliver banner pages as follows:

- Inserts a model data map (page format) or medium map (copy group) labeled \$BANNER—which exactly matches your banner pages—into the member in which your PAGEDEF and FORMDEF are defined
- Copies the first data or medium map into the same member and labels it \$\$FIRST (an update in place is achieved by using the same data set name for the old and new DDnames in the JCL for RMOPSF; a modified copy is created when the data set names differ)

\$BANNER and \$\$FIRST

\$BANNER and \$\$FIRST are not names of FORMDEF or PAGEDEF members, but internal labels in PAGEDEF and FORMDEF members.

Sample banner pages that contain these X'5A' records are provided in members TSTBNDL, TSTDIST, and TSTRPT in the library CAI.CVDED133.

X'5A' records and the \$BANNER and \$\$FIRST labels are included at the top and the bottom of each separator page between the statements /BEGSEP and /ENDSEP.

Using RMOPSF to Correct Banner Page Problems

To use RMOPSF to correct banner page problems, follow these steps:

1. Determine the PAGEDEF or FORMDEF resource you want to use as a model; ensure that it correctly formats the banner page you use.

Note: You can create a PAGEDEF or FORMDEF in a new member or use a PAGEDEF or FORMDEF defined in a member that IBM provides.

2. Prepare the JCL code that executes RMOPSF: include the MDLFDEF or MDLPDEF control statement to specify the name of the model you want to use, and include the SELECT control statement to prevent all members in the input library from being processed.
3. Execute the RMOPSF job.
4. Did you allocate the output library for RMOPSF to the AFP PSF task (does the new library data set name differ from the old library data set name)?
 - If no, go to step 5.
 - If yes, copy your modified PAGEDEF or FORMDEF resource to a library allocated to the AFP PSF.
5. Will you use the banners TSTBNDL, TSTDIST, and TSTRPT provided with CA Deliver?
 - If yes, go to step 6.
 - If no, insert the \$BANNER IMM and IDM at the top of your banner page, insert the \$\$FIRST IMM and IDM at the bottom of your banner page, and then load your new banners into the database with the RMODBASE utility BLOAD control statement.

Important! Do not insert \$BANNER and \$\$FIRST records into banner pages stored in AFP resources; instead, modify banner pages stored in the database.

Note: You do not need to recycle the started task to load these members.

6. Did you change or add the name of any member in CAI.CVDED133 when you loaded it?
 - If no, go to step 7.
 - If yes, enter the new name of the members in the BANNER field of the Report Definition Attributes online panel or the Bundle Definition Attributes online panel for the report or reports that use the modified PAGEDEF or FORMDEF.

Note: You can also specify the new names by resetting the initialization parameters BANNER, BNDLBNR1, BNDLBNR2, BNDLBNR3, and so on, but recycle the started task to reset initialization parameters.

7. Execute the job that generates your report.

When the report is printed by the AFP subsystem, the special commands in the banners switch to the data map or medium map with the \$BANNER label to print the start banner page, switch to the page format or overlay group with the \$\$FIRST label to print the report data, and then switch back to the \$BANNER label to print the end banner page.

Note: RMOPSF supports a wide range of page and form definition resources but not all. For example, page and form definition resources that contain conditional logic are not currently supported.

RMOPSF - Form and Page Definition Modification

The utility program RMOPSF in CAI.CVDELOAD modifies form and page definitions to allow banner pages to be printed with different characteristics than their corresponding reports when printed on an AFP page printer.

Examples of characteristics affected are notations and overlaps. This utility does this by taking an existing FORM or PAGE definition that contains a single medium map or data map and adds that medium map or data map to each FORM or PAGE definition selected by the utility.

Contents of FORMDEF and PAGEDEF

A FORMDEF consists of one or more medium maps and a PAGEDEF consists of one or more data maps. You can reference these maps by adding an Invoke Data Map (IDM) structured field or an Invoke Medium Map (IMM) structured field into your application program.

If IDM or IMM is omitted, the first map within the FORMDEF or PAGEDEF is automatically used.

Note: The IBM publication *PRINT SERVICES Facility Data Stream Reference* contains more information.

Example of Control Statements: ASCII

The following illustration depicts what the IMM and IDM control statements look like on the banner page in ASCII format:

```
***** TOP OF DATA *****
/BEGSEP
!..L....$BANNER          SET COPY-GROUP (INVOKE MEDIUM MAP)
!..L      $BANNER          SET PAGE-FORMAT (INVOKE DATA MAP)
1* START BUNDLE *****
* START BUNDLE ***** DELIVER
* START BUNDLE ***** REPORT DISTRIBUTION AND TRACKING
* START BUNDLE *****
* START BUNDLE ***** BUNDLE BANNER
* START BUNDLE *****
```

Example of Control Statements: Hexadecimal

The following illustration depicts what the IMM and IDM control statements look like on the banner page in hexadecimal format.

Job Control Statements

Specify the following JCL to execute RMOPSF:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMOPSF).

STEPLIB DD

Identifies the load library that contains RMOPSF.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

SYSUT1 DD

Defines the work disk space.

SYSUT2 DD

Defines the work disk space.

PDEFOLD DD

Defines the existing PAGE definition library.

PDEFNEW DD

Defines the converted PAGE definition library.

FDEFOLD DD

Defines the existing FORM definition library.

FDEFNEW DD

Defines the converted FORM definition library.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

MDLFDEF and MDLPDEF Control Statements

The MDLFDEF and MDLPDEF control statements are used to indicate the FORM or PAGE definition that contains the medium map or data map format used when printing the banner page. This medium map or data map is given the name \$BANNER and is inserted into the selected members. The model name is required only when you do not specify REMOVE; however, its use is recommended because the model name member is excluded from processing.

Syntax

```
MDLFDEF model
```

where *model* specifies the member of the FORM or PAGE definition library you want to use for printing the banner page. Copy the model from the "old" to the "new."

REMOVE Control Statement

The REMOVE control statement is used to indicate whether to add or remove the changes to the FORM or PAGE definition. If omitted, no is used by default.

Syntax

```
REMOVE option
```

where *option* specifies either yes (Y, remove) or no (N, add).

EXCLUDE Control Statement

The EXCLUDE control statement is used to exclude a definition from processing. Only one definition can be specified per EXCLUDE control statement; however, as many statements as necessary can be used. Precede the EXCLUDE statement by the model control statement or it is ignored.

SELECT Control Statement

The SELECT control statement is used to select a specific definition for processing; any other definition input to the utility but not also selected for processing is excluded from processing. Only one definition can be specified per SELECT control statement; however, as many statements as necessary can be used.

Example

The data set TEST.FDEFLIB.OLD contains a model form definition F1H10110 that is converted to a new name F1H10111 in TEST.FDEFLIB.NEW data set. The resulting FORMDEF resource needs to be copied to a library allocated to the AFP PSF.

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//RMOPSF EXEC PGM=RMOPSF
//STEPLIB DD DISP=SHR,DSN=CAI.CVDELOAD
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=&&WRK1,UNIT=SYSDA,SPACE=(TRK,(15,1))
//SYSUT2 DD DSN=&&WRK2,UNIT=SYSDA,SPACE=(TRK,(15,1))
//*PDEFOLD DD DSN=TEST.FDEFLIB.OLD,DISP=SHR
//*PDEFNEW DD DSN=TEST.FDEFLIB.NEW,DISP=SHR
//FDEFOLD DD DSN=TEST.FDEFLIB.OLD,DISP=SHR
//FDEFNEW DD DSN=TEST.FDEFLIB.NEW,DISP=SHR
//SYSIN DD *
MDLPDEF F1H10110
SELECT F1H10111
REMOVE NO
/*
//
```

RMORAP - Activating and Deactivating Reports from Batch

The RMORAP utility can be used to activate or deactivate reports from a batch job or started task. An *active report* is one that is scheduled to be produced during the current daily cycle. RMORAP is designed to accept input data directly from any of the job scheduling systems available.

Two types of data can be input to the utility:

- A list of job names
When given a list of job names as input, the program activates/deactivates all the reports produced by the specified jobs.
- A list of report identifiers
When given a list of report identifiers as input, the program activates/deactivates only those specific reports.

When both types of data are input to the program at the same time, both types of activation/deactivation are done. When a report to be bundled is activated/deactivated, the bundles for the report are also activated/deactivated. For RMORAP requests, the job does not complete until the actual update of the checkpoint has occurred.

Important! If you are using an authorization security exit or table and you try to activate/deactivate a report for which you are not authorized, the message "AUTHORIZATION FAILED" displays in the message area.

Reports can be automatically activated at the start of each new daily cycle through the RMORAP program by doing the following steps:

1. Create the start procedure JCL.
2. Specify the name of the start procedure with the START initialization parameter statement.

Create all input data to the procedure before you want to automatically start the procedure.

Important! The started task must be executing on the same operating system as all batch and online facilities that access checkpoint data detail and historical data, and facilities you use to delete definitions for these batch and online facilities to work.

Job Control Statements

Specify the following JCL to execute RMORAP:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMORAP) and, optionally, the high-level name of the database as the PARM parameter (PARM='DELIVER.SYSTEM1').

STEPLIB DD

Identifies the load library that contains RMORAP.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

JOBACT DD

Defines the sequential data set containing the names of the jobs whose reports are to be activated or deactivated.

The data set can contain fixed or variable records of any length. You can omit this DD statement if job names are not to be input to the program.

RPTACT DD

Defines the sequential data set containing the identifiers of the reports to be activated/deactivated.

The data set can contain fixed or variable records of any length. You can omit this DD statement if report identifiers are not to be input to the program.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

Important! Use either JOBACT DD or RPTACT DD but not both. RMORAP control statements must fit entirely on one card image. Continuations are not allowed.

JOB Control Statement

The JOB control statement is used to specify the format of the records containing the job names as input to the program. The default input format that is used if no control statement is provided is one job name per input record located in column 1 of the record.

Syntax

```
JOB nn1 nn2 ...
```

where *nn1* and *nn2* specify the position within the record (beginning with 1) of the job name. Multiple positions can be specified when more than one job name is contained in a record.

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database.

If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used.

REPORT Control Statement

The REPORT control statement is used to specify the format of the records containing the report identifiers as input to the program. The default input format that is used if no control statement is provided is one report identifier per input record located in column 1 of the record.

Syntax

```
REPORT nn1 nn2
```

where *nn1* and *nn2* specify the position within the record (beginning with 1) of the report identifier. Multiple positions can be specified when more than one report identifier is contained in a record.

TYPE Control Statement

The TYPE control statement is used to specify the type of processing to be performed, for example, report activation or deactivation. If omitted, report activation is performed.

Syntax

TYPE x

where x specifies a 1- or 2-character code:

- A = performs report activation
- U = performs report deactivation
- UF = forces report deactivation in all cases

Note: Use forced inactivation only as a last resort to clean up entries in the database; bundle holding copies can be left on the JES spool volumes.

Examples

The data set DELIVER.JOBACT contains a list of jobs with activated reports. The names of the jobs are located in positions 1, 21, and 41 of the records as follows:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//RAP      EXEC PGM=RMORAP
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//JOBACT   DD DSN=DELIVER.JOBACT,DISP=SHR
//SYSIN    DD *

NAME DELIVER.SYSTEM1
JOB 1 21 41
//
```

The following job deactivates the reports for the same list of jobs:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//RAP      EXEC PGM=RMORAP
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//JOBACT   DD DSN=DELIVER.JOBACT,DISP=SHR
//SYSIN    DD *
NAME DELIVER.SYSTEM1
TYPE U
JOB 1 21 41
//
```

RMORMS - Using Tracking

Use the RMORMS utility if you have CA 11 and want to use it with CA Deliver.

Once started, this batch utility program acts on the most recently produced reports. Reports on which you can act are as follows (you cannot act on reports that CA View has archived in the CA View database):

- Bundle holding copies in the JES spool
- Bundle holding copies placed in CA View database by a direct-to-View operation
- Non-bundled reports placed in the CA View database by a direct-to-View operation

Important! CA Deliver only backs out reports from the database specified in the SAR= initialization parameter.

If you use CA 11, execute RMORMS instead of UCC11RMS or U11RMS as the first step in your job.

Syntax

```
//step1 EXEC PGM=RMORMS,PARM='high-level-name,CA 11-parms'  
//RMONETn DD SYSOUT=x,DEST=dest,...,FREE=CLOSE  
//RMSPARMS DD DSN=name,DISP=SHR  
//SYSPRINT DD *
```

where *high-level-name* represents the name of the database.

Using AFP ACIF to Archive Reports

The CA View ACIF interface, which allows you to archive AFP reports, does not accept direct-to-View reports. You cannot, therefore, automatically back out direct-to-View AFP ACIF reports using the standard interface between CA Deliver and CA View.

For more information about setting up CA 11 (at a system level) to run with CA Deliver, see the *Installation Guide*.

Job Control Statements

Specify the following JCL to execute RMORMS:

EXEC

Specifies the program name (PGM=RMORMS) and the PARM parameter.

The PARM parameter specifies a single 1- to 17-character, high-level name of a database and the CA 11 program step parameters.

RMONETn DD

Specifies the destination to which the SYSOUT data set is sent.

The SYSOUT class, destination, and forms you specify must match the class, destination, and forms you specify for the SYSOUT that goes to the remote destination.

Do not specify this job control statement if CA 11 and CA Deliver are not executing on the same operating system.

RMSPARMS DD

Specifies the name of the data set in which the rerun control statements are to be input.

This job control statement is optional.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which the audit trail Rerun Processing Status report is to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

This job control statement is optional.

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database. You can specify NAME more than once.

FROM-NODE Control Statement

The FROM-NODE control statement is used to specify the name of the originating node on which CA 11 runs. Use NET-AND-LOCAL to indicate that your job outputs SYSOUT data sets to both the local and the remote CA Deliver.

Syntax

FROM-NODE *JES2-JES3-node-where-CA 11-runs* NET-AND-LOCAL

where *JES2-JES3-node-where-CA 11-runs* represents the name of the originating destination on which CA 11 runs.

LSERV-INBSSN Control Statement

The LSERV-INBSSN control statement is used to specify that CA 11 is to call the CA Balancing rerun interface routine. If omitted, this CA Deliver utility does not back out CA Balancing data for corresponding CA Deliver reports.

Rerun Processing Status Report

The RMORMS utility program, which allows you to use CA Deliver with CA 11, creates a status report at the end of the JES job log in the system on which CA Deliver executes. The Rerun Processing Status report provides you with an audit trail of rerun jobs.

04/28/2013 15:08:09 RMORMS		CA DELIVER RERUN PROCESSING STATUS			PAGE 1
JOBID	JOB STEP	PROC STEP	DD NAME	REPORT ID	STATUS
JOB05163	JSN1	PSN2	SYSUT2	ACCOUNT1	DELETED
JOB05163	JSN1	PSN2	SYSUT2	ACCOUNT2	FLAGGED
JOB05163	JSN2		SYSUT2	ACCOUNT3	FLAGGED
JOB05163	JSN2		SYSUT2	ACCOUNT3	FLAGGED
JOB05163	JSN3		SYSUT2	ACCOUNT4	SPOOL COPY ALREADY QUEUED, BUT SAR OR BUNDLE HOLDING COPY FLAGGED
JOB05163	JSN3		SYSPRINT	ACCOUNT5	SPOOL COPY ALREADY QUEUED, BUT SAR OR BUNDLE HOLDING COPY DELETED

Fields

The following are the fields on the Rerun Processing Status report and their descriptions:

JOBID

Specifies the number that JES assigns to the job.

JOB STEP

Specifies the name of the job step in the DD statement to which the report applies.

PROC STEP

Specifies the procedure step in which the DD statement for the report is found.

DD NAME

Specifies the 1–8 character name of the DD statement to which the report is written.

REPORT ID

Specifies the 1–32 character name that identifies the report.

STATUS

Provides the status of the report.

RMORPT - Batch Reporting

The RMORPT program is a general purpose reporting utility that provides hardcopy printout of information in the database. The RMORPT program provides the following functions:

- Report listing by job name
- Report listing by report identifier
- Report of distribution detail data by distribution identifier
- Report identifier cross-reference by distribution identifier
- Active report listing
- Active bundle listing
- Bundle listing by bundle identifier
- Banner page, printer setup, and online panel member listing

Job Control Statements

Specify the following JCL to execute RMORPT:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMORPT) and, optionally, the high-level name of the database as the PARM parameter (PARM='DELIVER. SYSTEM1').

STEPLIB DD

Identifies the load library that contains RMORPT.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

SORTLIB DD

Defines the load library containing the sort library programs.

This DD statement is only necessary when producing report 4.

SORTWK01 DD

Defines sort work disk space.

This DD statement is only necessary when producing report 4.

SORTWK02 DD

Defines sort work disk space.

This DD statement is only necessary when producing report 4.

SORTWK03 DD

Defines sort work disk space.

This DD statement is only necessary when producing report 4.

SYSOUT DD

Defines the sort message data set.

This DD statement is only necessary when producing report 4.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

NAME Control Statement

The NAME control statement is used to specify the high-level name of the database.

If omitted, the high-level name specified as the PARM parameter on the EXEC JCL statement is used. The NAME control statement applies to all control statements following it until another NAME control statement is encountered.

REPORT Control Statement

The REPORT control statement specifies the number of the report to be produced.

Syntax

```
REPORT nn
```

where *nn* specifies the number of the report (1–10) to be produced. This operand is required.

Report Listing by Job

Use the RMORPT utility REPORT 1 control statement to obtain a report listing by job. This report provides a listing of job name, step name, procedure step name, DDname, report identifier, and report type. Job names are printed in ascending sequence followed by the report identifiers. Each report identifier occupies a separate print line.

Report Listing by Report Identifier

Use the RMORPT utility REPORT 2 control statement to print a report listing by report identifier. This report provides a listing of report identifier, job name, step name, proc step, DDname, and distribution identifier. Report identifiers are printed in ascending sequence followed by the respective distribution identifiers. Grouped distribution identifiers appear horizontally on the print line with each new group starting on a new line.

Report of Distribution Detail Data by Distribution Identifier

Use the RMORPT utility REPORT 3 control statement to print a report listing of distribution detail data by distribution identifier. This report provides a listing of distribution identifier, destination, class, and distribution data. Distribution identifiers print in ascending sequence followed by the distribution detail lines (one per line).

Report Identifier Cross-Reference by Distribution Identifier

Use the RMORPT utility REPORT 4 control statement to print a listing of distribution identifiers and their related report identifiers. The distribution identifiers print in ascending sequence with the cross-referenced report identifiers printing horizontally on the page.

Active Report List

Use the RMORPT utility REPORT 5 control statement to print a listing of active reports. This report provides a listing of the active report identifiers and their job name, job ID, bundle identifier, external writer name of the bundle holding copy, system identifier, job status, and exception indicator. The listing appears in sequential order by report identifier.

Active Bundle List

Use the RMORPT utility REPORT 6 control statement to print a listing of active bundles. This report provides a listing of the active bundle identifiers and their system identifier, batch submission indicator, status, and exception indicator. The listing appears in sequential order by bundle identifier.

Bundle Listing by Bundle Identifier

Use the RMORPT utility REPORT 7 control statement to print a listing of bundle definitions by bundle identifier. This report provides a listing of bundle identifier, model banner page names, distribution identifier for the bundle, late time, bundling interval, submit indicator, report identifiers, and distribution identifiers.

Banner Page Member List

Use the RMORPT utility REPORT 8 control statement to print a listing of banner page members, including the date stored, time stored, and the number of records.

Printer Setup Member List

Use the RMORPT utility REPORT 9 control statement to print a listing of printer setup members, including the date stored, time stored, and the number of records.

Online Panel Member List

Use the RMORPT utility REPORT 10 control statement to print a listing of online panel members, including the date stored, time stored, and the number of records.

Examples

To produce a listing of each of the reports, execute the following job, which is located in RMORPT in CAI.CVDEJCL:

```
//EXAMPLE1 JOB ACCOUNT,PROGRAMMER
//REPORT EXEC PGM=RMORPT
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=BLKSIZE=133
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SYSOUT DD SYSOUT=A
//SYSIN DD *

NAME DELIVER.SYSTEM1
REPORT 1
REPORT 2
REPORT 3
REPORT 4
REPORT 5
REPORT 6
REPORT 7
REPORT 8
REPORT 9
REPORT 10
//
```

At the end of the current daily cycle, a hardcopy list of active reports and bundles is printed. Execute the previous job using report 5 and 6 only:

```
//EXAMPLE2 JOB ACCOUNT,PROGRAMMER
//LIST EXEC PGM=RMORPT
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=BLKSIZE=133
//SYSIN DD *
NAME DELIVER.SYSTEM1
REPORT 5
REPORT 6
//
```

RMORXB - Rebuilding Cross-Reference Records

RMORXB is a batch utility used to rebuild cross-reference records.

The database contains cross-reference records that maintain the relationship between distribution identifiers and reports, distribution identifiers and bundles, and distribution identifiers and distribution lists. On rare occasions, the cross-reference records can get out of sync due to abends or system outages. The RMORXB rebuilds these cross-reference records.

Job Control Statements

The following JCL is required to execute RMORXB:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMORXB) and the high-level name of the database as the PARM parameter (PARM='DELIVER.SYSTEM1').

STEPLIB DD

Defines the load library containing RMORXB.

If the program resides in a linklist library, omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which the report listing is to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

SORTLIB DD

Defines the load library containing the sort library programs.

SORTWK01 DD

Defines sort work disk space.

SORTWK02 DD

Defines sort work disk space.

SORTWK03 DD

Defines sort work disk space.

SYSOUT DD

Defines the sort message data set.

Example

To produce report cross-references for distribution identifiers, execute the following job which is located in RMORXB in CAI.CVDEJCL:

```
//EXAMPLE1 JOB ACCOUNT,PROGRAMMER
//STEP1 EXEC PGM=RMORXB,PARM='DELIVER.SYSTEM1'
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5))
//SYSOUT DD SYSOUT=A
//
```

RMOUTIL - Migration Support

The RMOUTIL program is used to copy, delete, or rename bundle, distribution, job, and report definitions. The utility was designed to simplify the process of migrating or propagating database records between databases.

Important! The started task must be executing on the same operating system as all batch and online facilities that access checkpoint data detail, historical data, and facilities you use to delete definitions for these batch and online facilities to work.

Rule for Updating Active Definitions

You can use RMOUTIL to update definitions while the started task is executing and accessing the database you specified for RMOUTIL. You cannot, however, update an active definition with RMOUTIL.

Do not update a definition when that definition is active (displayed as ACTIVE on the Report Panel) or is being used by an executing application job.

Job Control Statements

Specify the following JCL to execute RMOUTIL:

JOB

Initiates the job.

EXEC

Specifies the program name (PGM=RMOUTIL) and, optionally, the high-level name of the sending database at the PARM parameter (PARM='DELIVER.SYSTEM1').

STEPLIB DD

Identifies the load library that contains RMOUTIL.

If the program resides in a linklist library, you can omit this statement.

SYSPRINT DD

Identifies the sequential output data set (typically the SYSOUT) to which control statements and messages are to be sent.

If you do not specify a SYSOUT data set, specify DCB=BLKSIZE=nnn, where nnn represents a number that is a multiple of 133.

SYSIN DD

Specifies the name of the card image data set where the control statements you want to input are located.

COPY Control Statement

The COPY control statement is used to copy database definitions from a sending database to a receiving database. The COPY control statement supports the copying of bundle, distribution, job, and report data. All copied definitions maintain the original access information, last modified date, last modified time, and last user to modify. The sending database and receiving data are defined through the FROM and TO keywords of the NAME control statement, respectively.

Note: When using the external security with COPY, the TO database EXTSEC=YES parameter must be set and the external security is effective only after the CA Deliver started task is started or restarted.

Restrictions and Limitations

Due to the interconnections of database records and integrity of database operations, certain restrictions or limitations are imposed on the copy process.

Bundle records

If the bundle record previously exists on the receiving database, the bundle record is replaced.

Bundle distribution identifiers that do not exist in the receiving database are added to the receiving database.

Bundle report identifiers that do not exist in the receiving database are not copied (for example, the report identifiers are removed from the bundle record).

Distribution records

If the distribution record previously exists in the receiving database, the distribution record is replaced.

Job records

If the job record previously exists on the receiving database, the job record is replaced.

All reports defined to the job are copied (for example, added or replaced).

Report identifiers that are no longer defined to the job on the receiving database, if applicable, are deleted from the receiving database.

Report identifiers that are defined to another job on the receiving database, if the job previously exists, are not copied (for example, the report identifiers are removed from the copied job).

Distribution identifiers that are defined to reports within the job that do not exist in the receiving database are added to the receiving database.

Bundle references defined to reports within the job are not copied.

The bundle references that reside on the receiving database (if applicable) are maintained. The RMOUJR21 message identifies the bundle identifiers that existed on the sending database.

Report records

Report identifiers must be previously defined on the receiving database; this can be accomplished by copying the job definition first.

Bundle references defined to reports within the job are not copied.

The bundle references that reside on the receiving database (if applicable) are maintained. The RMOURR17 message identifies the bundle identifiers that existed on the sending database.

The report UNDEF is copied into the receiving database only if it is explicitly specified.

Syntax

```
/COPY BID=bundle-id  
      DISTID=distribution-id  
      JOB=jobname  
      RID=report-id
```

where:

bundle-id

Specifies the name of the bundle to be copied.

distribution-id

Specifies the name of the distribution identifier to be copied.

DISTID can be abbreviated as DID.

Note: If the Distribution-ID contains embedded blanks, enclose it in quotes (single or double). Any quote in Dist-ID must be entered as a pair of quotes because a non-paired quote ends the ID.

For example, if the value is JIM'S DESK, enter it as 'JIM'S DESK' or "JIM'S DESK".

jobname

Specifies the name of the job to be copied.

report-id

Specifies the name of the report identifier to be copied.

Note: You can use a wildcard to represent zero or more characters at the end of job names or bundle, distribution, or report identifiers. You cannot use the wildcard at the beginning or in the middle of job names or identifiers, but you can use it alone (for example, /COPY JOB=*, to copy all jobs).

The BID, DISTID, JOB, and RID keywords are mutually exclusive.

DELETE Control Statement

The DELETE control statement deletes database definitions from the sending database. The DELETE control statement supports the deleting of bundle, distribution, job, and report data. The sending database is defined using the FROM of the NAME control statement.

Syntax

```
/DELETE BID=bundle-id  
        DISTID=distribution-id  
        JOB=jobname  
        RID=report-id
```

where:

bundle-id

Specifies the name of the bundle to be deleted.

distribution-id

Specifies the name of the distribution identifier to be deleted.

DISTID can be abbreviated as DID.

Note: If the Distribution-ID contains embedded blanks, enclose it in quotes (single or double). Any quote in Dist-ID must be entered as a pair of quotes because a non-paired quote ends the ID.

For example, if the value is JIM'S DESK, enter it as 'JIM'S DESK' or "JIM'S DESK".

jobname

Specifies the name of the job to be deleted.

report-id

Specifies the name of the report identifier to be deleted.

Note: You can use an asterisk as a wildcard to represent zero or more characters at the end of job names or bundle, distribution, or report identifiers. You cannot use the wildcard at the beginning or in the middle of job names or identifiers, but you can use it alone (/DELETE JOB=*, to delete all jobs, for example).

The BID, DISTID, JOB, and RID keywords are mutually exclusive.

NAME Control Statement

The NAME control statement is used to define the sending database and/or the receiving database.

Syntax

```
/NAME FROM=sending-data-base  
      TO=receiving-data-base
```

where:

sending-data-base

Specifies the database high-level prefix of the sending database.

receiving-data-base

Specifies the database high-level prefix of the receiving database.

This parameter is only necessary for the COPY control statement.

RENAME Control Statement

The RENAME control statement renames database definitions in the sending database. The RENAME control statement supports the renaming of bundle, distribution, job, and report data. The sending database is defined through the FROM of the NAME control statement.

Syntax

```
/RENAME BID=bundle-id  
        DISTID=distribution-id  
        JOB=jobname  
        NEWNAME=newname  
        RID=report-id
```

where:

bundle-id

Specifies the name of the bundle to be renamed.

distribution-id

Specifies the name of the distribution identifier to be renamed.

DISTID can be abbreviated as DID.

Note: If the Distribution-ID contains embedded blanks, enclose it in quotes (single or double). Any quote in Dist-ID must be entered as a pair of quotes because a non-paired quote ends the ID.

For example, if the value is JIM'S DESK, enter it as 'JIM'S DESK' or "JIM'S DESK".

jobname

Specifies the name of the job to be renamed.

newname

Specifies the new name for the bundle, distribution, job, or report

Supply the new name that is different from the original.

report-id

Specifies the name of the report identifier to be renamed.

Note: The BID, DISTID, JOB, and RID keywords are mutually exclusive.

Examples

The following is an example of using RMOUTIL:

```
//EXAMPLE JOB MSGCLASS=T,NOTIFY=EXAMPLE,REGION=4096K
//STEP1 EXEC PGM=RMOUTIL
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

/NAME FROM=DELIVER.SYSTEM1
/DELETE JOB=JOB1
/RENAME JOB=JOB2 NEWNAME=JOB1
/NAME FROM=DELIVER.SYSTEM2 TO=DELIVER.SYSTEM1
/COPY JOB=JOB2
```

JOB1 is deleted from the DELIVER.SYSTEM1 database, and then JOB2 in DELIVER.SYSTEM1 is renamed to JOB1. Finally, the definition for job JOB2 from DELIVER.SYSTEM2 is copied to the definition of JOB2 of DELIVER.SYSTEM1.

The following is another example of using RMOUTIL:

```
//EXAMPLE JOB MSGCLASS=T,NOTIFY=EXAMPLE,REGION=4096K
//STEP1 EXEC PGM=RMOUTIL
//STEPLIB DD DSN=CAI.CVDELOAD,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

/NAME FROM=DELIVER.SYSTEM1
/DELETE JOB=PAY*
```

The job named PAY and all jobs that begin with the letters PAY are deleted from the DELIVER.SYSTEM1 database.

Chapter 5: The Database

This chapter covers these topics:

- Elements of the database and the utilities you use to initially build and modify data
- Types of data maintained in the database
- How to specify job name translation

This section contains the following topics:

[What is the Database?](#) (see page 271)

[Elements of the Database](#) (see page 271)

[Types of Data Maintained in the Database](#) (see page 275)

[Utilities for Building and Modifying the Database](#) (see page 276)

[Specifying Job Name Translation](#) (see page 277)

What is the Database?

The database is a set of one or more direct-access data sets that contain all of the data used by CA Deliver. The database is designed for high performance and quick access.

You create and maintain the database with the RMODBASE utility.

Note: This utility runs authorized.

Elements of the Database

This section tells you about the data sets and records that compose the database.

Rules and Guidelines

You can define a maximum of 127 data sets for the database. You can move data sets providing:

- The original and new volumes are of the same device type
- Each data set contains an equivalent amount of cylinders as the original,
- The new data sets each contain only one contiguous extent

Note: To improve performance, create a small number of large data sets rather than a large number of small data sets.

Database Data Set Attributes

The following attributes apply to the data sets in the database:

- DSORG=DA
- RECFM=F
- BLKSIZE=3768
- SPACE=(CYL,,,CONTIG)

Note: You can change the default block size (BLKSIZE) of 3768 with the RMODBASE utility, ADDDS control statement.

Syntax

index.RMODBASE.Dseq-number

where:

index

The high-level name of the database.

The high-level name is composed of one or more qualifiers separated by periods. You can specify an index that is from 1 to 17 characters long.

RMODBASE

A standard part of all database data set names.

RMODBASE does not represent variable text and does not change.

D

A data set type indicator which specifies that the current data set is a database component.

seq-number

The relative sequence number of the data set within the database.

The assigned seq-number is seven digits and can contain leading zeros.

Types of Records in the Database

The types of records in the database are:

Record Type	Abbreviation	Description
Master Control Record	MCR	Contains the control information, including the options that have been selected There is only one MCR.
Job Descriptor Record	JDR	Defines the DD statements within the job that produces the reports and identifies the reports There is one JDR for each job defined.
Report Descriptor Record	RDR	Defines the attributes, including distribution specifications and separation text, for a report There is one RDR for each report defined.
Bundle Descriptor Record (BDR)	BDR	Defines the contents of a bundle There is one BDR for each bundle defined.
Distribution Descriptor Record	DDR	Defines the address lines and destinations for a distribution identifier There is one DDR for each distribution identifier defined.
Report Historical Record	RHR	Defines the historical data for a report There is one RHR for each report for which historical data exists.
Bundle Historical Record	BHR	Defines the historical data for a bundle There is one BHR for each bundle for which historical data exists.
User Attribute Control Record	UCR	Describes the attributes for the online users (for example, PF key definitions) There is one UCR for each attribute for each user.
Online Member Control Record	OCR	Describes the panels, messages, and skeleton JCL members that are loaded from the online library There is one OCR for each panel, message, and skeleton member.

Record Type	Abbreviation	Description
Banner Member Control Record	BCR	Describes the model banner page members that are loaded from the model banner page library There is one BCR for each model banner page member.
Report Distribution Cross-Reference Record	XDR	Defines a distribution identifier and its cross-referenced reports
Bundle Distribution Cross-Reference Record	XBR	Defines a distribution identifier and its cross-referenced bundles
Distribution List Cross-Reference Record	XGR	Defines a distribution identifier and its cross-referenced distribution lists
Report Control Statements	RCS	Defines the attributes of a report definition

Note: Macros that map to the records (RMOMCR, RMOJDR, RMORDR/ RMORCS, RMOBDR, RMODDR, RMORHR/RMORHE, RMOBHR/RMOBHE, RMOUCR, RMOOCR, RMOBCR, RMOXDR, RMOXBR, and RMOXGR) are provided in CAI.CVDEMAG.

Structure of the Checkpoint Data Set

The checkpoint data set is divided into two sections of equal size.

Note: Be sure to allocate an even number of cylinders for the checkpoint data set):

- The first section holds actual checkpoint data.

The first four blocks of the checkpoint data section contain changed block masks for up to 32 different operating systems. These change masks enable each operating system to read only those blocks that have changed since the last time it accessed the checkpoint and to write only those blocks that it changes.

- The second section holds recovery data. This data is used if your system crashes while it is writing the changed blocks.

Data in the recovery section is recovered automatically.

Each section of the checkpoint data set can hold a maximum of 3,872 blocks.

Checkpoint Data Set Attributes

These attributes apply to the checkpoint data set in the database:

- DSORG=DA
- RECFM=F
- BLKSIZE=4096
- SPACE=(CYL,,,CONTIG)

Syntax

index.RMODBASE.Cseq-number

where:

RMODBASE

Defines a standard part of all database data set names assigned by CA Deliver.

RMODBASE does not represent variable text and does not change.

C

Specifies a data set type indicator (in this case, the current data set is a checkpoint data set).

C is assigned by CA Deliver.

seq-number

Specifies the relative sequence number of the data set within the database.

The assigned seq-number is seven digits and can contain leading zeros.

Types of Data Maintained in the Database

Report and bundle definitions and historical and tracking data are maintained in the database. In most cases, you use the online facility to access and change data in the database.

The types of data maintained in the database are:

- Job data -- Maintained in the database as individual entries identified by job name.
An entry describes the SYSOUT DD statements within the job that CA Deliver is to control.

- Report data -- Maintained as individual entries identified by a report identifier.

An entry does the following:

- Provides overrides to the JCL attributes for the report
- Assigns distribution identifiers to the report to identify the recipients of the report
- Defines the manner in which multiple reports written to the same DD statement are to be separated

- .Historical data -- Describes when the report was created; it is maintained for each report.

- Name, address, and destination data used to distribute reports to a user is maintained as individual entries identified by a distribution identifier.

- Bundle data is maintained as individual entries identified by a bundle identifier.

An entry identifies the distribution points and reports that are to be included in the bundle.

- Status information about the current and prior daily cycles is maintained for active reports.

An active report is one that you activate (that is, schedule to be created during the current cycle) or that CA Deliver automatically activates when the report is being created.

- Status information about the current and prior cycles is maintained for active bundles.

CA Deliver automatically activates a bundle whenever one of the reports defined for it is activated.

Utilities for Building and Modifying the Database

Three utilities are provided to aid you as you initially build the database and when you modify the definitions in the database from batch.

Note: For more information, see the chapter "Utilities."

- RMOJCL or RMOJCS scans the JCL for a job and automatically adds job and report description data to the database.

These utilities are interchangeable. The difference is that RMOJCS uses the CA JCLCheck Common Component.

After you execute either RMOJCL or RMOJCS, you can begin using CA Deliver to track and archive reports.

- RMODBB is a general purpose utility that adds and modifies the job, report, distribution, and bundle descriptions in the database.

This utility can extract the data to be added or modified from any sequential data set. The data set might be one that contains existing report descriptions or distribution specifications that you currently maintain or it might be a data set you create from existing data sets.

Specifying Job Name Translation

Use control statements, which are collectively referred to as the *job name translation table*, to translate the name of an executing job into another name.

The job name translation table is supplied by the started task. CA Deliver uses translated job names to search its database for the job definition data and report attribute definitions for a job.

If a job name is translated to a new name, only the translated name is used to search the database. A second search using the original name is not made when the translated name is not found.

You can, for example, run a daily, weekly, and monthly release of the same job under different job names. Because the jobs are identical in the reports they produce, and because the only essential difference between the jobs is their name, you need to define only one job for the database.

Use the DD statement RMOJTAB to input the control statements that compose the job name translation table.

Note: For more information about the operator command used to reset the job name translation table, see the chapter "Operator Commands."

Rules for Job Name Translation Control Statements

Keep these rules in mind when you specify job name translation control statements:

- Code each control statement as one card image in columns 1-71.
- Continuations are not allowed.

- Separate the two parameters that compose the control statement with one or more blanks or commas.
- The data set attributes must be sequential (DSORG=PS).

You can optionally use a member of a partitioned data set by referencing the member name on the DD statement for RMOJTAB.

Note: A partitioned data set with a member name allows you to update the job name translation table while the started task has the job name translation table allocated.

- Specify the RMOJTAB data set attributes as follows:

RECFM=FB, LRECL=80, BLKSIZE=xxxx

Set the block size to an amount that optimizes the space allocation for the DASD type at your site.

Syntax

executing-job-name translated-job-name

where:

translated-job-name

Specifies the translated job name of the job definition in the database.

You can specify an asterisk in any position in the job name to indicate that the corresponding character from the job name of the actual executing job is to be used.

The order of the control statements is significant:

- The first control statement that is found with an *executing-job-name* parameter that matches the job name of the actual, executing job is used.
- If no match is found in the table, the actual job name is used without translation.

Example

Assume the following job name translation table is defined to the started task:

```
A2765S5 A2765
PROD*      PRODJOB
*****D*****
*****W*****
*****M*****
```

These job names are going to be translated as specified when they are executed:

Execution Job Name	Translated Job Name
A2765S4	A2765S4
A2765S5	A2765
A2765S6	A2765S6
PROD6432	PRODJOB
PROD297J	PRODJOB
A2765S4D	A2765S4
B654R20M	B654R20
B654R20W	B654R20
B654R20J	B654R20J

Chapter 6: Model Banner Pages

This section describes model banner pages, which are distributed with CA Deliver. You create banner pages for reports and bundles based on the model banner page members you place in the model banner page library.

This section contains the following topics:

[Model Banner Page Library](#) (see page 281)

[Using Model Banner Page Members](#) (see page 281)

[Use Attribute Characters](#) (see page 287)

[Use Control Statements for Model Banner Page Members](#) (see page 288)

[Using Carriage Control Characters](#) (see page 290)

[Use Symbolic Variables](#) (see page 291)

Model Banner Page Library

The *model banner page library* is a partitioned data set that contains members that contain fixed, 133-byte records. Use these members as models to define the format, structure, and content of banner pages.

Load the members of the model banner page library into the database before you can use them. You can use these members to create new banner page members or modify the model banner page members distributed with CA Deliver.

Model banner page members are extracted when CA Deliver gains control of and creates the output for a SYSOUT data set.

Using Model Banner Page Members

A *model banner page member* is a member stored in the model library data set CAI.CVDED133, which is unloaded from the distribution tape during installation. You use model banner page members to define the format, structure, and content of banner pages.

Example 1

The following example displays the contents of a model banner page member:

```

/BEGSEP
1* START ***** START *
* START ***** AMALGAMATED AMERICAN MANUFACTURING ***** START *
* START ***** TEMPLE STREET FACILITY - LOS ANGELES, CA ***** START *
* START ***** START *
* &RID      &                                &*
*
*
*
*
*
*
*****
*
* REPORT ID: &RID      &   DATE: &DATE4 &
* JOBNAME:  &JNAME &     TIME: &TIME &
* JOBID:    &JID  &      CLASS: &C   &
*
*****
* SEND REPORTS TO:
*   &A11                      &
*   &A12                      &
*   &A13                      &
*   &A14                      &
*****
* SPECIAL INSTRUCTIONS:
*   &INST1                      &
*   &INST2                      &
*   &INST3                      &
*   &INST4                      &
*   &INST5                      &
*   &INST6                      &
*
*
*
*
* START ***** START *

```

Example 2

The following example displays another model banner page member:

```

/BEGSEP
1* START ***** START *
* START ***** AMALGAMATED ***** START *
* START ***** AMERICAN MANUFACTURING ***** START *
* START ***** TEMPLE STREET FACILITY ***** START *
* START ***** LOS ANGELES, CA ***** START *
* START ***** START *
*
* &RID & *
*
*
*
*
*
*
*
* &RID+16 & *
*
*
*
*
*
*
*
*
*
*****
*
* REPORT ID: &RID & PRINT DATE: &DATE4 & CREATE DATE: &CDATE4 &
* JOBNAME: &JNAME & PRINT TIME: &TIME & CREATE TIME: &CTIME &
* JOBJID: &JID & CLASS: &C &
*
*****
* SEND REPORTS TO:
*
* 1. &A11 &2. &A21 &
* &A12 & &A22 &
* &A13 & &A23 &
* &A14 & &A24 &
*
*****
* SPECIAL INSTRUCTIONS:
*
* &INST1 &
* &INST2 &
* &INST3 &
* &INST4 &
* &INST5 &
* &INST6 &
*
* START ***** START *
* START ***** START *

```


Example 4

The following banner page is produced from the model banner page member contents in Example 2.

```

1* START ***** START *
* START ***** START *
* START ***** AMALGAMATED ***** START *
* START ***** AMERICAN MANUFACTURING ***** START *
* START ***** TEMPLE STREET FACILITY ***** START *
* START ***** LOS ANGELES, CA ***** START *
* START ***** START *
*
* AAAAA CCCCC CCCCC 000000 U U N N TTTTTT SSSS PPPPP AAAAA Y Y AAAAA BBBB L EEEEE *
* A A C C C O O U U N N N T S S P P A A Y Y A A B B L E *
* A A C C C O O U U N N N T S S P P A A Y Y A A B B L E *
* AAAAA C C C O O U U N N N T SSSS PPPPP AAAAA Y AAAAA BBBB L EEEE *
* A A C C C O O U U N N N T S S P A A Y A A B B L E *
* A A C C C O O U U N N N T S S P A A Y A A B B L E *
* A A CCCCC CCCCC 000000 UUUU N N T SSSS P A A Y A A BBBB LLLLLL EEEEE *
*
* DDDDD AAAAA IIIII L Y Y RRRRR EEEEE PPPPP 000000 RRRRR TTTTT 0000 1 *
* D D A A I L Y Y R R E P P O O R R T 0 0 11 *
* D D A A I L Y Y R R E P P O O R R T 0 0 1 *
* D D AAAAA I L Y RRRRR EEEE PPPPP 0 0 RRRRR T 0 0 1 *
* D D A A I L Y R R E P 0 0 R R T 0 0 1 *
* D D A A I L Y R R E P 0 0 R R T 0 0 1 *
* DDDDD A A IIIII LLLLLL Y R R EEEEE P 000000 R R T 0000 1111 *
*
*****
*
* REPORT ID: ACCOUNTS_PAYABLE_DAILY_REPORT_01 PRINT DATE: 04/01/2013 CREATE DATE: 04/01/2013 *
* JOBNAME: APDAILY1 PRINT TIME: 16:11:57 CREATE TIME: 16:11:57 *
* JOBID: JOB01448 CLASS: W *
*
*****
*
* SEND REPORTS TO: *
*
* 1. DELIVER TO THE ACCOUNTS PAYABLE DEPARTMENT 2. *
*
*
*
*
*****
*
* SPECIAL INSTRUCTIONS: *
*
* CRITICAL REPORT - EXPEDITE *
* CRITICAL REPORT - EXPEDITE *
* CRITICAL REPORT - EXPEDITE *
* CRITICAL REPORT - EXPEDITE *
*
*
*
* START ***** START *
* START ***** START *

```

Types of Model Banner Page Members

You can use the following types of model banner page members:

Non-bundled report

Creates banner pages for a non-bundled report.

Separator

Creates separator pages in reports that contain control breaks.

Bundle

Creates the banner pages that are inserted before and after a complete bundle.

Distribution

Creates the banner pages that are inserted into a bundle before and after the complete set of reports for each distribution identifier.

Report

Creates the banner pages that are inserted into a bundle immediately before and after each report.

Specifying a Banner Page

Use the BANNER initialization parameter or BANNER field on the Report Definition Attributes panel to specify the name of the model banner page member you want to use for non-bundled reports.

Use the SEPARATOR field on the Control Break Identification sub-panel of the Report Definition Attributes panel to specify the name of the model banner page member you want to use for separator pages.

Use the BNDLBNR1, BNDLBNR2, and BNDLBNR3 initialization parameters or the BANNER field on the Bundle Definition Attributes panel to specify, respectively, the name of the model bundle, distribution, and bundled report banner page member you want to use in a bundle of reports.

Note: For more information about initialization parameters, see the chapter "Initialization Parameters." Online panels are described in the *Administration Guide*.

Use Attribute Characters

Use attribute characters to define the contents and layout of model banner pages. An *attribute character* is a special character you insert in the body of a model banner page member to define the beginning and end of symbolic variables and the location and size of text displayed on a banner page (symbolic variables are described later in this chapter).

The default attribute characters are described in the following table.

Note: You can change default attribute characters to characters you want to use with the /ATTR model banner page member control statement, which is described later.

Attribute Character	Symbol	Description
Ampersand	&	Used to define the start and end point of a symbolic variable you want inserted on a line on a banner page The first ampersand you insert defines the beginning of the symbolic variable. The second ampersand you insert defines the end of the symbolic variable. If you do not insert a second ampersand, the end of the line defines the end of the symbolic variable.
Percent symbol	%	Used to center text or a symbolic variable on a line Symbolic variables are inserted before they are centered.
Exclamation point	!	Used to convert text or a symbolic variable to large block letters on a line Large block letters measure 14 characters wide by 12 lines high.
Underscore	_	Used to convert text or a symbolic variable to small block letters on a line Small block letters measure eight characters wide by seven lines high.
Logical Not	¬	Used to convert text or a symbolic variable to large block letters and then centers the text or symbolic variable on a line Large block letters measure 14 characters wide by 12 lines high. Symbolic variables are inserted before they are centered.

Attribute Character	Symbol	Description
Cent symbol	¢	Used to convert text or a symbolic variable to small block letters and then centers the text or symbolic variable on a line Small block letters measure eight characters wide by seven lines high. Symbolic variables are inserted before they are centered.

Converted Block Letters

On the output banner page, converted block letters are output starting on the line that contains the attribute character and ending on subsequent lines. This means that you must leave enough blank lines to contain the block letters.

Substituting Text of Varying Lengths

If the length of text substituted for a symbolic variable exceeds the length you specify for the symbolic variable in the model banner page member, the substituted text is truncated.

If the length of text substituted for a symbolic variable is shorter than the length you specify in the model banner page member, the substituted text is padded with blanks at the end.

Use Control Statements for Model Banner Page Members

Use a set of control statements to define the contents, structure, and layout of model banner page members. Control statements are described in this section.

/BEGSEP Control Statement

The /BEGSEP control statement defines the start of a report banner page until the end of the member or until /END, /ENDSEP, or another /BEGSEP control statement is encountered. Insert the /BEGSEP control statement in column 1 of a record.

/ENDSEP Control Statement

The /ENDSEP control statement defines the end of a report banner page until the end of the member or until /END, /BEGSEP, or another /ENDSEP control statement is encountered. Insert the /ENDSEP control statement in column 1 of a record.

Use the /END control statement to terminate a banner page. Insert the /END control statement in column 1 of a record.

/ATTR Control Statement

Use the /ATTR control statement to change the default attribute characters to attribute characters you want.

The /ATTR control statement, which applies only to the banner page in which it is found, remains in effect until the end of the banner page member or until another /ATTR control statement is encountered.

Insert the /ATTR control statement in column 1 of a record.

Syntax

`/ATTR 123456`

where:

- 1 Represents the character you want to use instead of the ampersand (&), which you use to define the start and end points of a symbolic variable on a line
- 2 Represents the character you want to use instead of the percent symbol (%), which you use to center text or a symbolic variable on a line
- 3 Represents the character you want to use instead of the exclamation point (!), which you use to convert text or a symbolic variable to large block letters on a line
- 4 Represents the character you want to use instead of the underscore (_), which you use to convert text or a symbolic variable to small block letters on a line
- 5 Represents the character you want to use instead of the logical not (~), which you use to convert text or a symbolic variable to large block letters and then center the text or symbolic variable on a line
- 6 Represents the character you want to use instead of the cent symbol (¢), which you use to convert text or a symbolic variable to small block letters and then center the text or symbolic variable on a line

/ATTR Control Statement Syntax Rules

The following rules apply when specifying attribute characters with the /ATTR control statement:

- You must specify a character in position 1; you cannot specify a blank in position 1.
- To indicate that you do not want to change an attribute character, either insert a blank in positions 2–6, or insert the default attribute character in that position.
- You must insert a blank between /ATTR and the character in position 1.

Using Carriage Control Characters

The body of a banner page member comprises data lines. You specify carriage control in position 1 of a data line. Carriage control characters you can use in banner page members are discussed in this chapter.

Carriage Control on a Data Line

The following example displays a carriage control character on a data line:

```
1* START ***** START *
```

where *1*, in position 1 of this data line, is a carriage control character.

Types of Carriage Control Characters

The following carriage control characters can be used in a data line:

Carriage Control Character	Definition
1	Used to skip to channel 1 (a new page) and print the text on the line
2–9	Used to skip to channel 2, 3, 4, 5, 6, 7, 8, or 9 and print the text on the line
A, B, C	Used to skip to channel 10, 11, or 12 and prints the text on the line
+	Used to remain on the current line and print over the text on the current line
X'07'	Used to issue the special IBM 3800 printing subsystem "end of transmission" command

Carriage Control Character	Definition
X'17'	Used to issue the special IBM 3800 printing subsystem "mark form" command
X'5A'	Used to issue the special IBM 3800 printing subsystem "control record" command
J	Used in place of a 1 in the first record of a beginning banner page to force a skip to channel 1 when the banner is the first banner to be printed.
Blank or any character not listed in this table	Used to drop down one line on the output banner page and prints the text on the line

Rules for Carriage Control Characters

- You must specify carriage control characters in position 1 of a data line, and text and symbolic variables in positions 2–133.
- You can insert up to 974 data lines into the body of a banner page member.

Use Symbolic Variables

A *symbolic variable* is a software element capable of assuming a set of values. Use symbolic variables in model banner page members to identify the location where you want to insert a value or text.

Example

The following example displays a symbolic variable for which the SYSOUT destination of a report is inserted:

```
&DEST
```

Note: The ampersand indicates the start of a symbolic variable.

Types of Symbolic Variables

The following list describes symbolic variables:

Symbolic Variable	Abbrev.	Char. Length	Purpose
<i>Anm</i>	None	72	<p>Used to insert the one- to three- digit position number of the distribution identifier within a group of distribution identifiers (<i>n</i>), and the one-digit number of the address line of a distribution identifier (<i>m</i>)</p> <p>For example, A124 inserts address line 4 of the 12th distribution identifier in the group.</p> <p>For non-bundled report banner pages, the group comprises distribution identifiers grouped in the report definition.</p> <p>For bundled reports and bundle banner pages, the group comprises identifiers defined for the bundle in which reports actually exist.</p> <p>For distribution banner pages, the group comprises the single identifier for which reports are being printed.</p> <p>For report banner pages, the group comprises the single identifier for which the report is printed.</p> <p>If a group does not contain "<i>m</i>" identifiers, the value of this symbolic variable is blank.</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p>
ARCHID	None	32	<p>Used to insert the archival identifier name for the report</p> <p>If the archive identifier name is not specified in the report definition, the report identifier name is used.</p> <p>You can use this variable only on the report banner page and bundle report banner page. Blanks are substituted on the bundle banner page and bundle distribution banner page.</p>
<i>BAm</i>	None	72	<p>Used to insert the one-digit number of the address line of a distribution identifier (<i>m</i>) specified in the bundle definition</p> <p>Blank is substituted for this variable in non-bundled report banner pages.</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p>
BHDN	None	10	<p>Used to insert the history detail number of a bundle of reports, which remains the same for the entire bundle</p> <p>This variable does not work in banner pages for bundles that consist of reports that contain control breaks.</p>

Symbolic Variable	Abbrev.	Char. Length	Purpose
BID	None	32	<p>Used to insert the identifier for the bundle</p> <p>Blank is substituted for this variable in non-bundled report banner pages.</p> <p>You can also specify BID+nn to select the starting point for variable substitution. For example, BID+16 causes substitution to begin with the 17th character of the bundle identifier.</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p>
BSNO	None	5	<p>Used to insert the bundle sequence number</p> <p>Blank is substituted for this variable in non-bundled report banner pages.</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p>
CDATE	None	8	<p>Used to insert the creation date of the report or reports in the default date format with a two-digit year, for example, 10/01/02</p> <p>For non-bundled report banner pages, the creation date is identical to the print date except if the report is archived; in this case the creation date corresponds to the date the report was archived.</p> <p>For bundled reports and distribution and bundle banner pages, the creation date is identical to the print date.</p> <p>For report banner pages, the creation date is identical to the date the report bundle holding copy was created.</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p>
CDATE4	None	10	<p>Used to insert the creation date of the report or reports in the default date format with a four-digit year, for example, 10/01/2002</p> <p>For non-bundled report banner pages, the creation date is identical to the print date except if the report is archived; in this case the creation date corresponds to the date the report was archived.</p> <p>For bundled reports and distribution and bundle banner pages, the creation date is identical to the print date.</p> <p>For report banner pages, the creation date is identical to the date the report bundle holding copy was created.</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p>
CLASS	C	1	Used to insert the SYSOUT class of the report or reports

Symbolic Variable	Abbrev.	Char. Length	Purpose
COPIES	None	3	Used to insert the number of copies of the report For non-bundled report banner pages, the number of copies to be distributed is inserted. For bundled reports, 0 is inserted. This variable does not work in banner pages for reports that contain control breaks.
CTIME	None	8	Used to insert the time when a report is created in a <i>hh:mm:ss</i> format, for example, 10:45:59 For non-bundled report banner pages, the creation time is identical to the print time except if the report is archived; in this case the creation time corresponds to the time the report was archived. For bundled reports and distribution and bundle banner pages, the creation time is identical to the print time. For report banner pages, the creation date is identical to the time the report bundle holding copy was created. This variable does not work in banner pages for reports that contain control breaks.
CTIMEP	None	8	Used to insert the time when a report is created in a <i>hh.mm.ss</i> format, for example, 10.45.59 For non-bundled report banner pages, the creation time is identical to the print time except if the report is archived; in this case the creation time corresponds to the time the report was archived. For bundled reports and distribution and bundle banner pages, the creation time is identical to the print time. For report banner pages, the creation time is identical to the time the report bundle holding copy was created. This variable does not work in banner pages for reports that contain control breaks.
DATE	None	8	Used to insert the date when the report or reports are printed in the default date format with a two-digit year, for example, 10/01/02
DATE4	None	10	Used to insert the date when the report or reports are printed in the default date format with a four-digit year, for example, 10/01/2002
DDNAME	DD	8	Used to insert the DDname for the report or reports Blank is substituted for this variable in bundle and distribution banner pages.

Symbolic Variable	Abbrev.	Char. Length	Purpose
DESC	None	40	Used to insert the description of a report or bundle from the DESC field on either the Report or Bundle Definition Attributes panel Blank is substituted for this variable in bundle and distribution banner pages.
DEST	None	8	Used to insert the SYSOUT destination of a report or bundle
DHDN <i>n</i>	None	10	Used to insert the history detail number for a report within a group of reports for a specific distribution identifier, identified by <i>n</i> (a DHDN is created for each report in a bundle of reports) This variable does not work in banner pages for reports that contain control breaks. Note: Use this symbolic variable on report banner pages only.
DIST <i>n</i>	None	32	Inserts the one- to three-digit position number of the distribution identifier within a group of distribution identifiers (<i>n</i>) You need not specify the first distribution identifier; DIST and DIST1 are identical. For non-bundled report banner pages, the group comprises distribution identifiers grouped in the report definition. For bundled reports and bundle banner pages, the group comprises identifiers defined for the bundle in which reports actually exist. For distribution banner pages, the group comprises the single identifier for which reports are being printed. For report banner pages, the group comprises the single identifier of the printed report. You can also specify DIST+ <i>nn</i> to select the starting point for variable substitution. For example, DIST+16 causes substitution to begin with the 17th character of the bundle identifier. Note: This only works with the form DIST, not with DIST <i>n</i> .
DLVRID	None	32	Used to insert the Deliver identifier name for the report You can use this variable only on the report banner page and bundle report banner page. Blanks are substituted on the bundle banner page and bundle distribution banner page.
FORMS	None	8	Used to insert the name of the forms used

Symbolic Variable	Abbrev.	Char. Length	Purpose
HDN n	None	10	<p>Used to insert the HDN (history detail number) of each report in a bundle of reports identified by n</p> <p>An HDN is created for each report in a bundle of reports.</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p> <p>Note: Use this symbolic variable on report banner pages only.</p>
INST m	None	72	<p>Used to insert the one- to three-digit number of the instruction line (m) specified either on the Special Instructions sub-panel of the Report Definition Attributes panel for a report, or on the Special Instructions sub-panel of the Bundle Definition Attributes panel for a bundle</p> <p>You need not specify the first distribution identifier; INST and INST1 are identical.</p> <p>For distribution banner pages, the instructions printed are the instructions specified for the first report for the distribution identifier.</p> <p>If instruction lines are not defined, blank is inserted for this variable.</p>
JOBID	JID	8	<p>This variable specifies the subsystem job identifier of a report</p> <p>Blank is substituted for this variable in bundle and distribution banner pages.</p>
JOBNAME	None	8	<p>This variable specifies the job name for a report</p> <p>Blank is substituted for this variable in bundle and distribution banner pages.</p>
LINES	None	9	<p>This variable specifies the number of lines in a report, an entire bundle of reports, or all reports for a distribution identifier</p> <p>0 is inserted for this variable on start banner pages; the correct number of lines is inserted on end banner pages.</p>
NAME	None	8	<p>Used to insert the name of the model banner page member used as the basis for a banner page</p>
PAGES	None	12	<p>Used to insert the number of pages in a report, an entire bundle of reports, or all reports for a distribution identifier</p> <p>0 is inserted for this variable on start banner pages; the correct number of pages is inserted on end banner pages.</p>

Symbolic Variable	Abbrev.	Char. Length	Purpose
PREVRUN	None	6	<p>Used to insert the words RUN <i>nn</i>, where <i>nn</i> represents the number of times you have run the job during the current cycle</p> <p>PREVRUN is inserted only on banner pages of most recently run reports.</p> <p>Blank is substituted for this variable if a report is a non-bundled, spooled copy.</p> <p>Blank is substituted for this variable in bundle and distribution banner pages.</p> <p>Note: Use this symbolic variable on report banner pages only.</p>
PROCNAME	PN	8	<p>Used to insert the procedure step name for a report</p> <p>Blank is substituted for this variable in bundle and distribution banner pages.</p>
RERUN	None	9	<p>Used to insert the word CORRECTED only on banner pages of most recently rerun reports</p> <p>Blank is substituted for this variable if a report is a non-bundled, spooled copy.</p> <p>Blank is substituted for this variable in bundle and distribution banner pages.</p>
RHDN <i>n</i>	None	10	<p>Used to insert the report history detail number of a report based on its distribution identifier, identified by <i>n</i> (an RHDN is created for each report in a bundle of reports)</p> <p>This variable does not work in banner pages for reports that contain control breaks.</p> <p>Note: Use this variable on report banner pages only.</p>

Symbolic Variable	Abbrev.	Char. Length	Purpose
RID <i>n</i>	None	32	<p>Used to insert the one- to three-digit number of a report identifier in a group of report identifiers (<i>n</i>)</p> <p>You need not specify the first report identifier; RID and RID1 are identical.</p> <p>For non-bundled report banner pages, the group comprises the single report identifier being printed.</p> <p>For bundled reports and bundle banner pages, the group comprises all identifiers of the reports in the bundle that are actually being printed.</p> <p>For distribution banner pages, the group comprises all identifiers for reports that are actually being printed for the distribution identifier.</p> <p>For report banner pages, the group comprises the single identifier of the printed report.</p> <p>You can also specify RID+<i>nn</i> to select the starting point for variable substitution. For example, RID+16 causes substitution to begin with the 17th character of the bundle identifier.</p> <p>Note: This only works with the form RID, not with RID<i>n</i>.</p>
STEPNAME	SN	8	<p>Used to insert the step name for a report</p> <p>Blank is substituted for this variable in bundle and distribution banner pages.</p>
TIME	None	8	<p>Used to insert the time when a report is printed in a <i>hh:mm:ss</i> format, for example, 10:44:59</p>
TIMEP	None	8	<p>Used to insert the time when a report is printed in a <i>hh.mm.ss</i> format, for example, 10.44.59</p>
USER	None	20	<p>Used to insert the user accounting data for a report</p> <p>Blank is substituted for this variable in bundle and distribution banner pages.</p>
USERFLD <i>n</i>	None	Varies	<p>Used to insert the user field identification text extracted from a report that contains control breaks where <i>n</i> is a number from 1–9</p> <p>This value identifies the user field to be inserted on the Control Break Identification sub-panel on the Report Definition Attributes panel.</p>

Chapter 7: Setting Up Print Attributes

This section discusses how to define the printer setup member and tailor banner pages for printing on a Xerox 9700 printer including the following:

- Type 6 SMF (System Management Facilities) records
- Printer setup member
- Tailoring banner pages for printing on a Xerox 9700 printer

This section contains the following topics:

[Format of Type 6 SMF Records](#) (see page 299)

[Printer Setup Member](#) (see page 303)

[Tailor Banner Pages to Print on a Xerox 9700 Printer](#) (see page 304)

Format of Type 6 SMF Records

The user exit RMOSMFUX is invoked whenever CA Deliver writes a special type 6 SMF record. CA Deliver also generates special type 6 SMF records when you set the SMF initialization parameter to YES. A type 6 record is produced for each copy of a report.

This section describes the format of type 6 SMF records, which you must understand to use the RMOSMFUX user exit, which is described later in this chapter.

Note: For more information about the SMF initialization parameter, see the chapter "Initialization Parameters."

Example 1

DISTRIBUTION	COPIES
X	2
	>produces 3 records
Y	1

Example 2

```

DISTRIBUTION    COPIES
X                1
                >produces 2 records
Y                1

```

The format of type 6 SMF records is as follows:

Offset (dec.)	Name	Length	Description
0	SMF6LEN	2	Record length in binary
2	SMF6SEG	2	Segment descriptor
4	SMF6FLG	1	System indicator for VS2 (X'02')
5	SMF6RTY	1	Record type (X'06')
6	SMF6TME	4	Time in binary, in hundredths of a second, record was moved to SMF buffer
10	SMF6DTE	4	Date when record was moved to SMF buffer, in the format 00YYDDDF where F is the sign
14	SMF6SID	4	System identification
18	SMF6JBN	8	Job name
26	SMF6RST	4	Time in binary, in hundredths of a second, when reader recognized the job card for this job
30	SMF6RSD	4	Date when reader recognized the job card for this job, in the format 00YYDDDF where F is the sign
34	SMF6UIF	8	User identification
42	SMF6OWC	1	SYSOUT class
43	SMF6WST	4	Start time in binary, in hundredths of a second, when output queued to JES
47	SMF6WSD	4	Date output queued to JES, in the format 00YYDDDF where F is the sign
51	SMF6NLR	4	Number of lines, in binary, queued for printing
55	SMF6IOE	1	X'00'
56	SMF6NDS	1	Number of data sets (X'01')
57	SMF6FMN	4	Form name

Offset (dec.)	Name	Length	Description
61	SMF6PAD1	1	Section indicator: X'80' IBM 3800 printing subsystem section present X'40' Common section present
62	SMF6SBS	2	Subsystem identification (C'ED')
64	SMF6LN1	2	Length of rest of record, including this field, but not including any additional sections indicated by the SMF6PAD1 field
66	SMF6DCI	2	X'4000'
68	SMF6JNM	4	Job number
72	SMF6OUT	8	SYSOUT destination
80	SMF6FCB	4	FCB image identification
84	SMF6UCS	4	UCS image identification
88	SMF6PGE	4	Number of pages, in binary, queued for printing
92	SMF6RTE	2	X'0000'
94	SMF6RID	32	Report identifier
126	SMF6DID	32	Distribution identifier
158	SMF6BDLN	32	Bundle name
190	SMF6ACCT	20	Job accounting data

The format of type 6 SMF records (IBM 3800 printing subsystem section) is as follows:

Offset (dec.)	Name	Length	Description
+0	SMF6LN2	2	Length of IBM 3800 printing subsystem section, including this field
+2	SMF6CPS	8	Number of copies printed in each group Each byte represents, in binary, one copy group, and the sum of the 8 bytes is the total number of copies printed.
+10	SMF6CHR	16	Names of the character arrangement tables that define the characters used in printing Each name is 4 bytes long, with a maximum of four names.

Offset (dec.)	Name	Length	Description
+26	SMF6MID	4	Name of the copy modification module used to modify the data
+30	SMF6FLI	4	Name of the forms overlay printed on the copies
+34	SMF6FLC	1	Number of copies in binary on which the forms overlay is printed
+35	SMF6ID	1	Options indicator: X'80' Output is to be burst into sheets by the burster, trimmer, and stacker X'40' DCB sub-parameter OPTCD=J was specified

The format of type 6 SMF records (common section) is as follows:

Offset (dec.)	Name	Length	Description
+0	SMF6LN3	2	Length of common section, including this field
+2	SMF6ROUT	4	Output route code (not used)
+4	SMF6EFMN	8	Output forms name
+12		16	Not used
+28	SMF6JBID	8	Job number
+36	SMF6STNM	8	Step name
+44	SMF6PRNM	8	Procedure step name
+52	SMF6DDNM	8	DDname
+60	SMF6USID	8	User ID
+68	SMF6SECS	8	Security label (not used)
+72	SMF6PRMD	8	Processing mode (not used)
+80	SMF6DSNM	53	Data set resource name (not used)
+133		3	Not used
+136	SMF6OTOK	20	Output group token (not used)

Printer Setup Member

You can use printer setup members to set printer attributes. This section describes printer setup members.

The *printer setup library* is a partitioned data set that comprises members that contain fixed, 133-byte records. Once you create the data set, you can add members to, or modify members in, the data set.

A *printer setup member* is a member in which you set printer attributes. This member comprises records of SYSOUT data.

Format of the Records in a Printer Setup Member

The first position in each record in a printer setup member contains a valid American Standards Association (ASA) carriage control character.

The format of the records in a printer setup member depends on the requirements of your printer. To print on a Xerox 9700 (or compatible) printer, for example, the records in the printer setup member can be DJDE statements. To print on an IBM 3800 (or compatible) printer on which the Print Services Facility (PSF) is used, the records can be X'5A' control records.

Note: You can insert only 50 records in a printer setup member.

Loading Printer Setup Library Members to the Database

Load the members of the printer setup library into the database before they can be used.

To add printer setup members to the database, use the PLOAD function of the RMODBASE utility, which is described in the chapter "Utilities."

Appending Printer Setup Records to a Report

To append printer setup records to a report, specify the name of a printer setup member in the PRSET field on the Report Definition Attributes panel. PRSET is described in the *Administration Guide*.

Printer setup records are appended to a report immediately after the report banner page but before the first record in a report.

If you do not specify the name of a printer setup member on the Report Definition Attributes panel, printer setup records are not appended to the report. A default printer setup member name is not used.

Tailor Banner Pages to Print on a Xerox 9700 Printer

Reports printed with CA Deliver (including reprints from CA View) can be tailored for the Xerox 9700 printer by generating DJDE in the banner pages for reports.

Tailoring Model Banner Pages

Because page-oriented DJDE commands are only effective at the next page boundary, you may need to insert a skip-to-channel-1 carriage control on the next model banner page line after the DJDE statements.

You need not insert a skip-to-channel-1 carriage control when the DJDE statements are specified as the last lines in a model banner page, because the report that follows always begins on a new page boundary.

- When you define a model banner page to be used for reports, you code the DJDE statements as either constant lines of data to be used for all reports or as variable lines of data that can be changed in each report definition.
- To specify constant DJDE data in a model banner page, code the DJDE statements as one or more lines in the model banner page.
- To specify variable DJDE data in a model banner page, code one or more lines in the model banner page as instruction lines using an &INSTn symbolic variable.
- When you define a report that is to use the model banner page, you specify the DJDE statements for the report as special instructions on the Report Definition Attributes panel.

Example

Assume the following options:

- Banner pages are always printed with one logical page per physical page.
- Reports can be printed with multiple logical pages on one physical page; the actual format is specified as a special instruction in the report definition.
- The following Page Descriptor Entries (PDEs) have been created:
 - PDE1 One logical page per physical page
 - PDE4 Four logical pages per physical page
- The DJDE offset is 0 and the identifying prefix is \$\$DJDE.

The beginning model banner page is set up as follows:

```
/BEGSEP
  $$DJDE FORMAT=PDE1,END;
1
...body of beginning model banner page...

&INST1
```

The ending model banner page is set up as follows:

```
/ENDSEP
  $$DJDE FORMAT=PDE1,END;
1
...body of ending model banner page...
```

For four pages per physical page, you can specify the following statement on the first line of the Special Instructions sub-panel of the Report Definition Attributes panel:

```
$$DJDE FORMAT=PDE4,END;
```

DJDE Identification

DJDE records start at column position printer offset +2 in banner pages.

Optional Use of PRSET

You can optionally implement the previous DJDE statements using the PRSET field on the Report Definition Attributes panel. In PRSET, the column position is printer offset +1.

Tailoring Bundle Model Banner Pages

You can specify DJDE statements in bundle model banner pages similar to report model banner pages. In this case, however, you can specify variable DJDE statements as special instructions on the Bundle Definition Attributes panel rather than on the Report Definition Attributes panel.

Note: Check your installation standards for DJDE offset and prefix, as they differ from printer to printer.

Chapter 8: Accessing Programs from Global Subsystem

This section discusses the following topics:

- How to establish a host command environment in CA GSS (Global Subsystem)
- Commands processed by the host command environment
- GREXX variables
- A REXX EXEC that illustrates the use of the *DELIVER* host command environment

This section contains the following topics:

[Establishing a Host Command Environment](#) (see page 307)

Establishing a Host Command Environment

You can use CA Deliver and CA GSS together to establish a host command environment in CA GSS for the CA Deliver programs RMODBASE, RMODBB, and RMOGRW. The host command environment is named DELIVER and is provided by the module RMOINTF. It is accessed by the *ADDRESS DELIVER* GREXX instruction.

Note: For more information about the host command environment, see the *CA Common Services for z/OS Getting Started*.

For more information about the instructions for installing the DELIVER host command environment into CA GSS, see the *Installation Guide*.

Commands Processed by the Host Command

The commands processed by the DELIVER host command environment are the same as those that can be specified on the SYSIN control statements that are processed by RMODBASE, RMODBB, and /REPORT.

Note: /REPORT causes RMOGRW to be invoked.

GREXX Variables

The following GREXX variables are used in the host command environment:

- **XPDELIVER.DBASE**

This variable contains the CA Deliver database prefix that is typically specified in the NAME *high-level-name* control statement for the RMODBASE utility.

Specify XPDELIVER.DBASE before any CA Deliver commands are issued. The length of the prefix must be less than or equal to 17.

- **RC**

The DELIVER host command environment upon completion of the requested command sets this variable.

RC contains the return code from the requested command as follows:

Return Code	Description
-3	Command not found
-2	Not enough memory to perform command
-1	Unable to access shared variable pool
0	OK
4	Warning message issued—message text on stack
8	Error message issued—message text on stack
9	XPDELIVER.DBASE not set
10	XPDELIVER.DBASE string too long
12	Severe error message issued—message text on stack
16	Fatal error message issued—message text on stack
28	Language processor environment could not be found (could be a CA GSS installation problem)
32	Internal error
Other	Internal error

Example of a REXX EXEC

The following REXX EXEC example illustrates the use of the DELIVER host command environment:

```
#DESC TEST OF THE DELIVER HOST COMMAND ENVIRONMENT
#SOURCE
/*:RMOGRW.TEST1
/CONTROL DATABASE=DELIVER.SYSTEM1
/OUTPUT DISTID COL(1)
/OUTPUT DDEST COL(35)
/DEFINE (I) BIN
/DO I=1 TO NA BY 1
/  OUTPUT A COL(55)
/  NEXT A
/END

*/
TRACE OFF
XPDELIVER.DBASE=DELIVER.SYSTEM1'
CALL QUEUEPGM('RMOGRW.TEST1')
ADDRESS DELIVER '/REPORT'
NUM = QUEUED()
RECORD.0 = NUM
DO I = 1 TO NUM
  PULL RECORD.I
END
TRACE ALL
DDNAME = ALLOC(, 'NEW DELETE', 'PERM', 'DATA',,,, 'TRK 10',,
               'PS', 'FB', '256', '2560')
SAY DDNAME
DCB = SAM('OBTAIN', 'LOCAL', 'DATA', 'OUTPUT')
SAY DCB
X = SAM('OPEN', DCB)
SAY X
DODBB = 'NO'
DO I = 1 TO RECORD.0
  A1 = SUBSTR(RECORD.I, 30, 10)
  IF A1 = 'JIM SMITH' THEN DO
    OUTRECORD = 'D' || SUBSTR(RECORD.I, 1, 8)
    X = SAM('PUT', DCB, OUTRECORD, ' ')
    SAY OUTRECORD
    DODBB='YES'
  ELSE DO
    SAY 'SKIPPING' RECORD.I
  END
END
```

```

(BLOCK TEXT C)  END
X = SAM('CLOSE', DCB)
IF DODBB = 'YES' THEN DO
    ADDRESS DELIVER '/DISTDEF FUNCTION=(1) DISTID=(3)'
    SAY RC
    CALL PRINTRESULT
END
X = DEALLOC ('DATA')
ADDRESS DELIVER 'ADDS CYLINDER=3'
CALL PRINTRESULT
EXIT

QUEUEPGM:
    PARSE ARG NAME
    KEY = '/*: ' || NAME
    KEYL = LENGTH(KEY)
    LINENUM = 1
    DONE = 0
    DO WHILE DONE=0
        LINE = SOURCELINE(LINENUM)
        LINENUM = LINENUM + 1
        IF SUBSTR(LINE,1,KEYL) = KEY THEN DO
            DO WHILE DONE=0
                LINE = SOURCELINE(LINENUM)
                LINENUM = LINENUM + 1
                IF SUBSTR(LINE,1,2) = '*/' THEN DO
                    DONE = 1
                END
            ELSE DO
                QUEUE LINE
            END
        END
    END
    ELSE IF SUBSTR(LINE,1,8) = 'QUEUEPGM' THEN DO
        DONE = 1
    END
RETURN

PRINTRESULT:
    HOWMANY = QUEUED()
    SAY HOWMANY
    DO I = 1 TO HOWMANY
        PULL RECORD
        SAY RECORD
    END
RETURN

```

Chapter 9: NJE Unattended Download

This section discusses the unattended download feature and covers special software requirements, and how to send reports or bundles to CA Output Management Document Viewer (CA OM Document Viewer).

This section contains the following topics:

[Overview of Unattended Downloading](#) (see page 311)

Overview of Unattended Downloading

The NJE unattended download feature allows you to request the unattended download of reports and bundles from CA Deliver to an NJE node on the LAN that is serving PCs running CA Output Management Document Viewer. This feature uses CA Connect to manage the NJE print routing on the LAN.

This feature provides support for unattended download capability to CA OM Document Viewer from the open front end, batch bundle print function, and started task bundle print function.

Unattended downloading provides the following benefits:

- You do not have to wait for a terminal-based transfer to complete when transferring SYSOUT to CA OM Document Viewer.
- Report bundles can be delivered directly from CA Deliver to CA OM Document Viewer.

Software Requirements

The unattended download feature requires the following software:

- MVS operating system Release 5 or higher
- JES2 Release 5 or higher or JES3 Release 5 or higher
- CA Connect Release 1.01 or higher
- CA Output Management Document Viewer

Note: The unattended download feature does not download the ACIF-generated index for AFP reports.

Sending a Report or Bundle for Document Viewing

To send a report or bundle to CA OM Document Viewer using CA Connect, set up a DISTID for the CA Connect destination.

On the Distribution Data for ID panel, the DEST field must contain the NJE NODEID.USERID that CA Connect has designated for CA OM Document Viewer, and set the CA Connect Node field to YES.

Note: For more information about sending a report, see your CA OM Document Viewer or CA Connect documentation.

For more information about online panels, see the *Administration Guide*.

Chapter 10: Security

This section explains the internal and external security features including the following topics:

- Internal security
 - Types and levels of security access
 - Macros you use to define authorization criteria in the security table
 - Syntax of the RMOAGRP macro
 - Syntax of the RMOATH macro
 - Syntax of the RMOAEND macro
 - Coding macros
- External security
 - Types of resources protected
 - Levels of security access
 - How to implement external security for CA Top Secret Security (formerly known as eTrust CA-Top Secret), CA ACF2 Security (formerly known as eTrust CA-ACF2), and the IBM RACF product

This section contains the following topics:

[Internal Security](#) (see page 313)

[External Security](#) (see page 326)

Internal Security

This section discusses the database and the access levels that are managed by security table RMOATHTB or, optionally, the authorization security user exit RMOATHUX.

The AUTHTID initialization parameter can be used to identify a unique security table other than default security table, RMOATHTB. However, the RMODBASE utility always uses the default security table (RMOATHTB).

Initial authorization checking verifies your user ID against CA Deliver's security table. If there is no security table, all users are granted full access. Then, the security user exit checks your user ID. This user exit (which you write) can override the access authority assigned by the security table.

If there is no security user exit, the security table authorization remains in effect. If the security table or the security user exit is not present, all users have full access.

You can define multiple authorization tables.

Important! Define only one table per database. If you bring up two CA Deliver tasks with the same database and specify different tables for each, after the second task is started, both tasks use the table specified in the second task.

RMOATHTB Security

The types and levels of security access that are available are as follows:

Security Level	Type of Access
ACT	Updates report activation data You can activate or inactivate and issue U, UF, and F tabular commands on the Active Report List panel. You can issue the A tabular command on the Job or Report Selection List panel.
BACT	Updates bundle activation data You can issue F and P tabular commands on the Active Bundle List panel.
BUNDLE	Bundle definitions
DBASE	Database using the RMODBASE utility, which runs authorized
DIST	Distribution definitions
JOB	Job definitions
REPORT	Report definitions
BANNER	Banner pages
PANEL	Panel members

The available levels of access are as follows:

BROWSE

Browses the database.

UPDATE

Browses and updates the database.

DELETE

Deletes members from the database.

RENAME

Renames members on the database.

OPERATOR

Operator functions such as activation and immediate bundle printing.

ADMIN

All of the above.

Required Access Levels

The following list identifies the required access level to perform the function:

BROWSE

Browses definitions, displays active reports, displays history, displays job data, displays cross reference.

UPDATE

All BROWSE functions and the ability to update the information presented.

DELETE

Deletes definitions.

RENAME

Renames definitions.

OPERATOR

Activates, deactivates, forced deactivates, frees all, prints bundle now.

ADMIN

All of the above.

Using Macros to Define Authorization Criteria

A set of macros enables you to define your authorization criteria for use by the security table. The macro instructions are as follows:

RMOAGRP

Defines a group of users to be given authorization.

RMOATH

Defines authority for one or more groups of users.

RMOAEND

Completes the security table.

The macros are coded in standard assembler language syntax.

Macro Coding Order

The order in which the macros are coded is fixed. Code the security table as follows:

1. Specify one or more RMOAGRP macros to define the groups of users.
Note: The order of the RMOAGRP macros is significant in that the first group found that matches the user's attributes or environment is the group that is used.
2. Specify one or more RMOATH macros to define all the authorization criteria.
3. Specify the RMOAEND macro to complete the security table.
4. Specify the END assembler language statement.

RMOAGRP Macro

Syntax

```
name RMOAGRP    USER=(userid-list),      X
              ACCOUNT=x,                  X
              OPER=x,                      X
              BATCH=x,                    X
              STC=x,                      X
              LIMIT=x,                    X
              EXCL=x
```

Where:

USER=(userid-list)

Specifies a list of generic user identifiers defining the list of users contained in the group.

If omitted, USER=* (all user identifiers) is assumed.

ACCOUNT=x[NO, YES]

Specifies whether the users specified must also be TSO users with account authority.

If omitted, ACCOUNT=NO is assumed.

Note: Some security systems are restricted to 7-character user IDs. Be sure that your user IDs (including the mask) conform to the restrictions of your security package.

OPER=x[NO, YES]

Specifies whether the users specified must also be TSO users with operator authority.

If omitted, OPER=NO is assumed.

BATCH=x[NO, YES]

Specifies whether a batch job for which no user identifier is supplied on the JOB JCL statement is a member of the group.

If omitted, BATCH=NO is assumed.

This operand is used as an alternative to the USER, ACCOUNT, and OPER operands.

For an online user or batch job/started task with a user identifier specified on its JOB JCL statement, the USER, ACCOUNT, and OPER operands are used exclusively to determine the group.

For a batch job with no user identifier specified on its JOB JCL statement, the BATCH operand is used exclusively to determine the group.

STC=x[NO, YES]

Specifies whether a started task for which no user identifier is supplied on the JOB JCL statement is a member of the group.

If omitted, STC=NO is assumed.

This operand is used as an alternative to the USER, ACCOUNT, and OPER operands.

For an online user or batch job/started task with a user identifier specified on its JOB JCL statement, the USER, ACCOUNT, and OPER operands are used exclusively to determine the group.

For a started task with no user identifier specified on its JOB JCL statement, the STC operand is used exclusively to determine the group.

LIMIT=x[NO, YES]

Specifies whether lists displayed contain only information the user is authorized to see (YES) or contain information the user is not authorized to see (NO).

This applies to selection lists and lists of items associated with the current subject.

For example, LIMIT=YES suppresses display of certain reports from the Reports For Job XYZ panel. Additional overhead is involved if LIMIT=YES is specified because each item in the list must be passed through the security process.

LIMIT=NO does not entail this individual item checking.

If omitted, LIMIT=NO is assumed.

Note: All continuation lines (USER, ACCOUNT, OPER, BATCH, STC, LIMIT) must begin in column 16; column 72 of the preceding record must contain a non-blank character.

EXCL=x[NO, YES]

Indicates to exclude this entry from "online" candidacy.

Authorization Group Criteria

The following table provides the criteria necessary for a particular type of program (online, batch, started task) to be considered as a member of an authorization group:

Program Type	Could Be Part of Group if	Not Part of Group if
Online	User IDs match and, if table has OPER=YES, TSO must have operator authority, or User IDs match and, if table has ACCOUNT=YES, TSO must have account authority	User IDs match and EXCL=YES
Batch job without user ID	Table has BATCH=YES	
Batch job with user ID	User IDs match, and table has BATCH=YES	Table has either ACCOUNT=YES or OPER=YES
Started task without user ID	Table has STC=YES	
Started task with user ID	User IDs match, and table has STC=YES	Table has either ACCOUNT=YES or OPER=YES

RMOATH Macro

Syntax

```
label RMOATH type,           X
      ID=(identifier-list),  X
      ACCESS=(access-list),  X
      GROUP=(group-list),    X
      PROMPT=x
```

Where:

label Not used

type Specifies the type of access this macro applies

Valid types are:

ACT Updates report activation data (activate or inactivate and Free tabular command F on the Active Report List panel)

BACT Updates bundle activation data (Free tabular command F and Print Bundle Now tabular command P on the Active Bundle List panel)

BUNDLE Bundle definitions

DBASE Database using the RMODBASE utility, which runs authorized

DIST Distribution definitions

JOB Job definitions

REPORT Report definitions

BALANCE Balancing option

BANNER Panel

ID= Specifies a list of generic identifiers defining the list of identifiers for which the groups are authorized

(*identifier-list*)

Types of identifiers correspond to the types of access as follows:

ACT Report identifiers

BACT Bundle identifiers

BUNDLE Bundle identifiers

DBASE Not applicable

DIST Distribution identifiers

JOB Job names

REPORT Report identifiers

If omitted, ID=* (all identifiers) is assumed

ACCESS=
(*access-list*) Specifies a list of access levels for which the groups are authorized as follows:

BROWSE Browses the database

UPDATE Browses and updates the database

DELETE Deletes members from the database

RENAME Renames members on the database

OPERATOR Operator functions such as activation, immediate bundle printing, and so on

ADMIN All of the above

GROUP=
(*group-list*) Specifies the list of groups of users that are authorized for the type of access

The names of the groups must be identical to those specified on the RMOAGRP macros; generic group names cannot be used.

PROMPT=x
[*NO*, *YES*] Specifies whether the operator is prompted to authorize access
If omitted, PROMPT=*NO* is assumed.

Note: All continuation lines (type, ID, ACCESS, GROUP, PROMPT) must begin in column 16; column 72 of the preceding record must contain a non-blank character.

RMOAEND Macro

Syntax

RMOAEND

Coding Macros

When coding the macros, you can specify *generic identifiers*. For the purposes of these macros only, you can code a generic identifier as a combination of the actual characters of the identifier and asterisks (*). For any position of the identifier, except for the last that an asterisk is coded, an exact match of that one position is not required. When the last position of the generic identifier contains an asterisk, any number of characters can follow the identifier.

For example, the following table lists generic and actual identifiers and specifies whether the generic and actual identifiers match:

Generic Identifier	Actual Identifier	Match
TECHVS	TECHVS	Yes
TECHVS	TECHVSE	No
TECH*	TECHVS	Yes
T*CHVS	TECHVS	Yes
T*CHVS	TACHVS	Yes
T*CHVS	TEACHVS	No
A*B*C*	AABBCCDD	Yes
A*B*C	AABBCCDD	No
A*B*C	AYBYC	Yes

After you code the macros, assemble and link edit the security table as RMOATHTB with authorization code 1 and then place the security table in the library that contains CA Deliver's load modules.

Important! When coding RMOAGRP USER=, the value you specify for USER= is the identifier assigned to the executing job or task by your external security system.

Example

The following code enables the started task and RMORAP to activate bundles:

```
actgroup RMOAGRP USER=(RMOSTC,RMORAP),          X
          STC=YES,                                X
          BATCH=YES
          ...
          RMOATH BACT,                             X
          ID=*,                                    X
          ACCESS=(ADMIN),                          X
          GROUP=(actgroup)
          ...
          RMOATH ACT,                              X
          ID=*,                                    X
          ACCESS=(ADMIN),                          X
          GROUP=(actgroup)
          ...
          RMOAEND
          END
```

Note: All continuation lines must begin in column 16; column 72 of the preceding record must contain a non-blank character.

Example

The following code is an example of a complete RMOATHTB. This code is located in CAI.CVDEOPTN as RMOATH1. An additional sample, RMOATH2, which authorizes all users to all functions, can also be found in CAI.CVDEOPTN. You can tailor SMP USERMOD HBBOATH and can use it to install the tables.

```
*****
*          THIS IS A SAMPLE RMOATHTB TABLE FOR USE          *
*                                                                *
*          WITH CA DELIVER                                     *
*                                                                *
*                                                                *
*  AFTER THE MACROS HAVE BEEN CODED THE SECURITY TABLE MUST BE  *
*  ASSEMBLED WITH ASSEMBLER H USING THE ALIGN OPTION. IT IS THEN *
*  LINKED AS RMOATHTB WITH OPTIONS AC=1 AND RENT. PLACE THE LOAD *
*  MODULE IN THE AUTHORIZED LIBRARY CONTAINING THE CA DELIVER    *
*  LOAD MODULES.                                                 *
```

```
*
* THIS IS AN ASSEMBLER LANGUAGE PROGRAM. TO CONTINUE A
* STATEMENT ON THE NEXT LINE YOU MUST END THE CURRENT LINE
* WITH A COMMA, PLACE A NON-BLANK CHARACTER IN 72, AND CONTINUE
* IN COLUMN 16 ON THE NEXT LINE. AN ASTERISK IN COLUMN 1
* DENOTES A COMMENT.
*
*****

*
* FIRST WE DEFINE FOUR GROUPS
*
* NOTES
*   - IF AUTOACT=YES, THE USERID ASSOCIATED WITH THE
*     CA DELIVER STARTED TASK MUST BELONG TO A SECURITY GROUP
*     THAT HAS ACTIVATE AUTHORITY FOR REPORTS AND BUNDLES.
*
*
*   - IF START=RMORAP, THE USERID ASSOCIATED WITH THE RMORAP
*     STARTED TASK MUST BELONG TO A SECURITY GROUP THAT HAS
*     ACTIVATE AUTHORITY FOR REPORTS AND BUNDLES.
*
*
*   - IF RMORAP IS RUN IN BATCH, THE USERID ASSOCIATED WITH
*     THE BATCH JOB MUST BELONG TO A SECURITY GROUP THAT HAS
*     ACTIVATE AUTHORITY FOR REPORTS AND BUNDLES AND HAS
*     "BATCH=YES".
*
*     BATCH=YES INDICATES THAT THE USERIDS IN THIS GROUP HAVE
*     AUTHORITY TO UPDATE THE DATABASE FROM A BATCH JOB (I.E.
*     RMODBB, RMORAP ...)
```

```
*
*
* STC=YES INDICATES THAT ONE OR MORE USERIDS IN THIS GROUP
* ARE STARTED TASKS.
*
*
* - USER=(-),BATCH=YES,STC=YES WILL ONLY ASSIGN STARTED
*   TASKS AND BATCH JOBS WITH A BLANK (UNASSIGNED) USERID
*   TO THIS SECURITY GROUP
```

```
*
*   - USER=(*) ,BATCH=YES,STC=YES WILL ASSIGN EVERY USER
*     TO THIS SECURITY GROUP IN ADDITION TO ALL BATCH JOBS
*     AND STARTED TASKS WITH A BLANK (UNASSIGNED) USERID.
*
*   - IF THE USER PARAMETER IS OMITTED, USER=(*) IS ASSUMED.
*

SYSADMIN RMOAGRP USER=(BMCLAUG,GHASUL,JMOONEY,TBURNET,TFLEMIN,    X
          JBUTLER,EHART,TCROSSL) ,BATCH=YES
APPLPROG RMOAGRP USER=(GDS,BBB) ,LIMIT=YES
CLSONLY  RMOAGRP USER=(GDT)
ACTGROUP RMOAGRP USER=(RMOSTC,RMORAP) ,STC=YES,BATCH=YES
* HERE'S WHAT SYSADMIN CAN DO (ANYTHING)
* NOTE - ACCESS=(ADMIN) IS THE SAME AS
*        ACCESS=(UPDATE,RENAME,DELETE,OPERATOR)
*
*        BACT AUTHORITY OF OPERATOR ALLOWS USERS TO
*        ISSUE THE P COMMAND AND UPDATE
*        DETAIL HISTORY FROM THE ACTIVE BUNDLE SCREEN.
*
*        ACT AUTHORITY OF OPERATOR ALLOWS USERS TO
*        ISSUE THE U, F, AND UF COMMANDS AND UPDATE
*        DETAIL HISTORY FROM THE ACTIVE REPORT SCREEN.
*
*        BUNDLE AUTHORITY OF OPERATOR ALLOWS USERS TO UPDATE
*        DETAIL HISTORY FROM THE BUNDLE SELECTION LIST.
*
*        REPORT AUTHORITY OF OPERATOR ALLOWS USERS TO UPDATE
*        DETAIL HISTORY FROM THE REPORT SELECTION LIST.
*
* DEFAULT VALUES:
*   ID=(*) ,ACCESS=(ADMIN)
*
RMOATH ACT ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH BACT ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH BUNDLE ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH DBASE ,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH DIST ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH JOB ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH REPORT ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH BANNER ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
RMOATH PANEL ,ID=*,ACCESS=(ADMIN) ,GROUP=(SYSADMIN)
*
```

```

* HERE'S WHAT APPLICATION PROGRAMMING CAN DO
*
* NOTE - IF GROUP-1 USERS CAN UPDATE A REPORT AND
*         GROUP-2 USERS CAN DELETE A REPORT
*         THEN A USER BELONGING TO BOTH GROUPS CAN UPDATE AND DELETE.
*
RMOATH ACT, ID=(TST*),                                X
        ACCESS=(OPERATOR), GROUP=(APPLPROG)
RMOATH BACT, ID=(TST*),                                X
        ACCESS=(OPERATOR), GROUP=(APPLPROG)
RMOATH BUNDLE, ID=(TST*),                              X
        ACCESS=(UPDATE, DELETE, RENAME), GROUP=(APPLPROG)
RMOATH DIST, ID=(TST*),                                X
        ACCESS=(UPDATE, DELETE, RENAME), GROUP=(APPLPROG)
RMOATH JOB, ID=(TST*),                                  X
        ACCESS=(ADMIN), GROUP=(APPLPROG)
RMOATH REPORT, ID=(TST*),                              X
        ACCESS=(ADMIN), GROUP=(APPLPROG)
*
* HERE'S WHAT CLSONLY CAN DO
*
RMOATH ACT, ID=CLS*, ACCESS=(BROWSE), GROUP=(CLSONLY)
RMOATH BACT, ID=CLS*, ACCESS=(BROWSE), GROUP=(CLSONLY)
RMOATH BUNDLE, ID=CLS*, ACCESS=(BROWSE), GROUP=(CLSONLY)
RMOATH DIST, ID=CLS*, ACCESS=(BROWSE), GROUP=(CLSONLY)
RMOATH JOB, ID=CLS*, ACCESS=(BROWSE), GROUP=(CLSONLY)
RMOATH REPORT, ID=CLS*, ACCESS=(BROWSE), GROUP=(CLSONLY)
*
* HERE'S WHAT ACTGROUP (STARTED TASKS AND BATCH JOBS) CAN DO
*
RMOATH ACT, ID=*, ACCESS=(OPERATOR), GROUP=(ACTGROUP)
RMOATH BACT, ID=*, ACCESS=(OPERATOR), GROUP=(ACTGROUP)
* THE NEXT 6 STATEMENTS ARE FOR USERS THAT DO NOT HAVE A SECURITY
* SYSTEM INSTALLED BUT WANT TO UPDATE THE DATABASE IN BATCH
* RMOATH BUNDLE, ID=*, ACCESS=(ADMIN), GROUP=(ACTGROUP)
* RMOATH DBASE, ACCESS=(ADMIN), GROUP=(ACTGROUP)
* RMOATH DIST, ID=*, ACCESS=(ADMIN), GROUP=(ACTGROUP)
* RMOATH JOB, ID=*, ACCESS=(ADMIN), GROUP=(ACTGROUP)
* RMOATH REPORT, ID=*, ACCESS=(ADMIN), GROUP=(ACTGROUP)
*
* COMPLETE THE SECURITY TABLE
RMOAEND
END

```

External Security

The sections that follow demonstrate how to implement external security for the following security managers:

- CA Top Secret
- CA ACF2
- IBM's RACF

Because of the way that CA Deliver handles security, you cannot see any violations logged by the external security manager. CA Deliver preapproves users for access to resources. That is, before displaying a panel, CA Deliver checks for everything a user is allowed to do or see on that panel.

It is impossible for a violation to occur in this environment and, therefore, none are logged. For example, if a user who is authorized to the Report function requests a list of reports, the only reports displayed are those permitted for that user. If no reports are permitted, an empty list is displayed.

Note: Because the CA Deliver external security processing verifies the user's authorization for every object requested, you can minimize the impact on performance by using a wildcard character to request only specific data. For example, if you enter J to display all jobs, performance is impacted. You can improve performance by entering J ABC* to view only those jobs that begin with ABC.

The implementation procedures presented in this section are examples only. You must determine the appropriate settings for your environment.

Important! These examples authorize all users to do everything. We recommend that the CA Deliver administrator work together with a security administrator to do the implementation as a cooperative effort.

Initialization Parameters

Three initialization parameters affect the operation of external security.

- EXTSEC=

Set EXTSEC=NO so that external security calls are not made.

You can also code EXTSEC=YES and EXTSEC=UNIQUE to cause CA Deliver to make external security calls. The difference is in the resource names that are used. For EXTSEC=YES or EXTSEC=UNIQUE to be in effect, the started task must have been started at least once after the last IPL.

All CA Deliver resource names are prefixed with either "RMO." or "RMO#." If you specify EXTSEC=YES, the "RMO." prefix is used. Use this option when all (or most) of your CA Deliver regions use the same security rules. For more information about resource names, see the section Resources and Authorization that follows.

User ID Determination

CA Deliver uses exits to determine the user ID of the user accessing the CA Deliver database. These exits can also perform initial logon functions for users that are logging on using the VTAM and CICS XMS interfaces.

Note: For more information about these exits, see the RMOUSxUX - User ID Determination Exits in the chapter "User Exits" in the *Programming Guide*.

■ EXTTRAN=

If you use external security (RACF, CA ACF2 Security, CA Top Secret Security) and the report id contains characters from the extended special character set, this parameter must be specified as EXTTRAN=YES to have extended special characters automatically translated to '_' underscores, before the RACROUTE security call.

Note: For more information about this parameter, click [Character Translation in Resources Names](#) (see page 328).

■ SYSID=

If you need separate rules for one or more regions, code EXTSEC=UNIQUE in the initialization parameters. External security calls then use the "RMO#." prefix. The #-sign is replaced with the region's SYSID= value.

Note: For more information about initialization parameters, see the chapter "Initialization Parameters."

Resources and Authorizations

CA Deliver manages security with nine resource types. Each type is given a name containing a national character to satisfy the requirements of the various security packages. Each resource type corresponds to a TYPE from the RMOATHTB internal security table as shown in the following table.

Resource Type	RMOATHTB Type	ACF2 Type	Resources Protected
DLV@ACT	ACT	DAC	Activation, inactivation, and updates on the Active Reports panel
DLV@BACT	BACT	DBA	Updates on the Active Bundles panel
DLV@BANR	BANNER	DBR	Displaying and deleting banners (DIS B)

Resource Type	RMOATHTB Type	ACF2 Type	Resources Protected
DLV@BNDL	BUNDLE	DBN	Bundles
DLV@DBAS	DBASE	DBS	RMODBASE actions
DLV@DIST	DIST	DDI	Distribution IDs
DLV@JOB	JOB	DJB	Jobs
DLV@PANL	PANEL	DPN	Displaying and deleting panels (DIS O)
DLV@REPT	REPORT	DRP	Reports

Character Translation in Resource Names

Certain characters which are allowed in report names and other definitions can be treated as 'wildcards' by some security products.

If the following character appears in a resource name, it is translated to the character specified in all resource names.

- All Resource Names:

Character	Translated to
& (ampersand)	! (exclamation)
* (asterisk)	+ (plus)
% (percent)	(bar)

- Report Resource Names and Distribution IDs:

Character	Translated to
' ' (blank)	_ (underscore)
& (ampersand)	! (exclamation)
* (asterisk)	+ (plus)
% (percent)	(bar)

- If **EXTTRAN=YES** specified - Report Resource Names and Distribution IDs:

Character	Translated to
& (ampersand)	! (exclamation)
* (asterisk)	+ (plus)

Character	Translated to
% (percent)	(bar)
' ' (blank)	_ (underscore)
¢ (cent sign)	_ (underscore)
! (exclamation point)	_ (underscore)
/ (slash)	_ (underscore)
< (less than)	_ (underscore)
((left parentheses)	_ (underscore)
(bar)	_ (underscore)
) (right parentheses)	_ (underscore)
; (semicolon)	_ (underscore)
~ (not sign)	_ (underscore)
¡ (broken bar)	_ (underscore)
, (comma)	_ (underscore)
> (greater than)	_ (underscore)
? (question mark)	_ (underscore)
: (colon)	_ (underscore)
' (single quote)	_ (underscore)
= (equal sign)	_ (underscore)
" (double quote)	_ (underscore)

Access Levels

The internal security of CA Deliver allows six different access levels. To be compatible with the external security managers, compress these into four levels of access. They are inclusive in that a higher access level implies all lower levels.

Note: Because of the nature of CA Deliver's SAF calls, this is true even when using CA ACF2.

RMOATHTB Access	RACF	TSS	ACF2	Description
BROWSE	READ	READ	READ	Browses the database
UPDATE	UPDATE	UPDATE	UPDATE	Browses and updates the database

RMOATHTB Access	RACF	TSS	ACF2	Description
OPERATOR and UPDATE	CONTROL	CONTROL	DELETE	Provides operator functions such as activation and immediate bundle printing
DELETE	ALTER	ALL	ADD	Deletes members from the database
RENAME	ALTER	ALL	ADD	Renames members on the database
ADMIN	ALTER	ALL	ADD	Provides all of the above

As mentioned previously, resource names are prefixed with either "RMO." or "RMO#." What follows the prefix depends on the type of resource.

Resource Type	Data Type	Resource Name	Function
DLV@ACT	Report	RMO(#{reportid	A – Active Display
DLV@BACT	Bundle	RMO(#{bundle	A – Active Display
DLV@BANR	Banner	RMO(#{banner	N/A
DLV@BNDL	Bundle	RMO(#{bundle	B – Bundle Data
DLV@DBAS	Database	RMO(#{dbhlq	N/A
DLV@DIST	Distribution ID	RMO(#{distid	D – Distribution Data
DLV@JOB	Job	RMO(#{job	J – Job Data
DLV@PANL	Panel	RMO(#{panel	N/A
DLV@REPT	Report	RMO(#{reportid	R – Report Data

You can use generics. For example, "RMO.*" covers every entity of a given type.

There is a special case for each resource type except DLV@BANR, DLV@DBAS and DLV@PANL. To perform a given function at all, the user must have at least READ access to a resource named "RMO." or "RMO#" for each function. The most visible effect of this is on the CA Deliver Primary Selection panel. A user who, for example, does not have READ access to "RMO." in type DLV@REPT, do not even have the R (Report Data) option available. Defining this as a generic resource works but, gives read access to every resource of that type. To prevent this, grant READ access to a non-generic resource: "RMO." instead of "RMO.*" or "RMO.(G)".

The following sections detail the steps necessary to implement support of external security with CA Deliver. There are descriptions and sample jobs for CA Top Secret, CA ACF2, and IBM's RACF.

Implementing External Security for CA Top Secret

To implement external security for CA Top Secret, do the following:

Note: For more information about the commands listed in this section, see the *CA Top Secret Command Functions Guide*. The sample jobs can be found in CVDEJCL member RMOTSS.

1. Add the CA Deliver resource types (classes) to the Resource Descriptor Table, for example:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//RDT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
```

TSS ADDTO(RDT) RESCLASS(DLV@ACT) RESCODE(37)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	
TSS ADDTO(RDT) RESCLASS(DLV@BACT) RESCODE(38)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	
TSS ADDTO(RDT) RESCLASS(DLV@BANR) RESCODE(39)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	
TSS ADDTO(RDT) RESCLASS(DLV@BNDL) RESCODE(3A)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	
TSS ADDTO(RDT) RESCLASS(DLV@DBAS) RESCODE(3B)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	
TSS ADDTO(RDT) RESCLASS(DLV@DIST) RESCODE(3C)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	
TSS ADDTO(RDT) RESCLASS(DLV@JOB) RESCODE(3D)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	
TSS ADDTO(RDT) RESCLASS(DLV@PANL) RESCODE(3E)	+
ATTR(LONG,NONGENERIC)	+
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +	
DEFACC(NONE)	

```
TSS ADDTO(RDT) RESCLASS(DLV@REPT) RESCODE(3F)      +
ATTR(LONG,NONGENERIC)                               +
ACLST(ALL,CONTROL,UPDATE,READ,NONE) +
DEFACC(NONE)
/*
```

2. Create a department to own the resources, for example:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//DEPT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
TSS CREATE(DLVRDEPT) TYPE(DEPT) NAME('DELIVER DEPARTMENT')
/*
```

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//OWNER EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
TSS ADDTO(DLVRDEPT) DLV@ACT(RMO.)
TSS ADDTO(DLVRDEPT) DLV@BACT(RMO.)
TSS ADDTO(DLVRDEPT) DLV@BANR(RMO.)
TSS ADDTO(DLVRDEPT) DLV@BNDL(RMO.)
TSS ADDTO(DLVRDEPT) DLV@DBAS(RMO.)
TSS ADDTO(DLVRDEPT) DLV@DIST(RMO.)
TSS ADDTO(DLVRDEPT) DLV@JOB(RMO.)
TSS ADDTO(DLVRDEPT) DLV@PANL(RMO.)
TSS ADDTO(DLVRDEPT) DLV@REPT(RMO.)
/*
```

3. Make a profile and permit resource access to it, for example:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//PROFILE EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
TSS CREATE(DLVRPROF) TYPE(PROFILE) NAME('DELIVER') DEPT(DLVRDEPT)
/*
```

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//PERMIT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
TSS PERMIT(DLVRPROF) DLV@ACT(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
TSS PERMIT(DLVRPROF) DLV@BACT(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
TSS PERMIT(DLVRPROF) DLV@BANR(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
TSS PERMIT(DLVRPROF) DLV@BNDL(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
```

```

TSS PERMIT(DLVRPROF) DLV@DBAS(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
TSS PERMIT(DLVRPROF) DLV@DIST(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
TSS PERMIT(DLVRPROF) DLV@JOB(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
TSS PERMIT(DLVRPROF) DLV@PANL(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
TSS PERMIT(DLVRPROF) DLV@REPT(RMO.(G)) ACCESS(ALL) ACTION(FAIL)
/*

```

4. Add the profile to a user, for example:

```

//EXAMPLE JOB ACCOUNT,PROGRAMMER
//ADDTO EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
TSS ADDTO(userid) PROFILE(DLVRPROF)
/*

```

Implement External Security for CA ACF2

To implement external security for CA ACF2, do the following:

Note: For more information about the commands listed here, see the *CA ACF2 Administration Guide*. The sample jobs can be found in CVDEJCL member RMOACF2.

1. Map the CA Deliver resource types to CA ACF2 resource types, for example:

```

//EXAMPLE JOB ACCOUNT,PROGRAMMER
//CLAS EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *

ACF
SET CONTROL(GSO)
INS CLASMAP.DLV@ACT RESOURCE(DLV@ACT) RSRCTYPE(DAC) ENTITYLN(37)
INS CLASMAP.DLV@BACT RESOURCE(DLV@BACT) RSRCTYPE(DBA) ENTITYLN(37)
INS CLASMAP.DLV@BANR RESOURCE(DLV@BANR) RSRCTYPE(DBR) ENTITYLN(13)
INS CLASMAP.DLV@BNDL RESOURCE(DLV@BNDL) RSRCTYPE(DBN) ENTITYLN(37)
INS CLASMAP.DLV@DBAS RESOURCE(DLV@DBAS) RSRCTYPE(DBS) ENTITYLN(22)
INS CLASMAP.DLV@DIST RESOURCE(DLV@DIST) RSRCTYPE(DDI) ENTITYLN(37)
INS CLASMAP.DLV@JOB RESOURCE(DLV@JOB) RSRCTYPE(DJB) ENTITYLN(13)
INS CLASMAP.DLV@PANL RESOURCE(DLV@PANL) RSRCTYPE(DPN) ENTITYLN(13)
INS CLASMAP.DLV@REPT RESOURCE(DLV@REPT) RSRCTYPE(DRP) ENTITYLN(37)
/*

```

2. Tell CA ACF2 about the SAF calls that CA Deliver is making, for example:

```

//EXAMPLE JOB ACCOUNT,PROGRAMMER
//SAFD EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *

```

```
ACF
SET CONTROL(GS0)
INS SAFDEF.DLV@ACT ID(DLV@ACT) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@ACT,STATUS=ACCESS)
INS SAFDEF.DLV@BACT ID(DLV@BACT) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@BACT,STATUS=ACCESS)
INS SAFDEF.DLV@BANR ID(DLV@BANR) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@BANR,STATUS=ACCESS)
INS SAFDEF.DLV@BNDL ID(DLV@BNDL) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@BNDL,STATUS=ACCESS)
INS SAFDEF.DLV@DBAS ID(DLV@DBAS) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@DBAS,STATUS=ACCESS)

INS SAFDEF.DLV@DIST ID(DLV@DIST) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@DIST,STATUS=ACCESS)
INS SAFDEF.DLV@JOB ID(DLV@JOB) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@JOB,STATUS=ACCESS)
INS SAFDEF.DLV@PANL ID(DLV@PANL) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@PANL,STATUS=ACCESS)
INS SAFDEF.DLV@REPT ID(DLV@REPT) PROGRAM(RMO-) RB(RMO-) -
    NOAPFCHK RACROUTE (REQUEST=AUTH,CLASS=DLV@REPT,STATUS=ACCESS)
/*
```

3. Make the resource types resident, for example:

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//ACF2 EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
```

```
ACF
SET CONTROL(GS0)
CHANGE INFODIR TYPES(R-RDAC,R-RDBA,R-RDBR,R-RDBN,R-RDBS)
CHANGE INFODIR TYPES(R-RDDI,R-RDJB,R-RDPN,R-RDRP)
/*
```

4. Enter the modify console commands to refresh all of this, for example:

```
F ACF2,REFRESH(CLASMAP)
F ACF2,REFRESH(SAFDEF)
F ACF2,REFRESH(INFODIR)
```

5. Define CA ACF2 rules, for example:

Note: The rule definitions used in the following example are contained in nine separate members of a PDS, called *RULES.PDS*. For more information about PDS, see PDS Members.

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//RULE EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
```

```
ACF
SET RESOURCE(DAC)
COMPILE 'RULES.PDS(DAC)'
STORE
SET RESOURCE(DBA)
COMPILE 'RULES.PDS(DBA)'
STORE
SET RESOURCE(DBN)
COMPILE 'RULES.PDS(DBN)'
STORE
SET RESOURCE(DBR)
COMPILE 'RULES.PDS(DBR)'
STORE
SET RESOURCE(DBS)
COMPILE 'RULES.PDS(DBS)'
STORE
SET RESOURCE(DDI)
COMPILE 'RULES.PDS(DDI)'
STORE
SET RESOURCE(DJB)
COMPILE 'RULES.PDS(DJB)'
```

```
STORE
SET RESOURCE(DPN)
COMPILE 'RULES.PDS(DPN)'
STORE
SET RESOURCE(DRP)
COMPILE 'RULES.PDS(DRP)'
STORE
/*
```

6. Tell CA ACF2 to rebuild the resident rules, for example:

```
F ACF2,REBUILD(DAC)
F ACF2,REBUILD(DBA)
F ACF2,REBUILD(DBR)
F ACF2,REBUILD(DBN)
F ACF2,REBUILD(DBS)
F ACF2,REBUILD(DDI)
F ACF2,REBUILD(DJB)
F ACF2,REBUILD(DPN)
F ACF2,REBUILD(DRP)
```

PDS Members

Following are the PDS members.

Member	Contents
DAC	* DLV@ACT REPORT ACTIVATION \$KEY(RMO) TYPE(DAC) UID(*) SERVICE(READ) ALLOW - UID(*) SERVICE(ADD) ALLOW
DBA	* DLV@BACT BUNDLE ACTIVATION \$KEY(RMO) TYPE(DBA) UID(*) SERVICE(READ) ALLOW - UID(*) SERVICE(ADD) ALLOW
DBN	* DLV@BNDL BUNDLES \$KEY(RMO) TYPE(DBN) UID(*) SERVICE(READ) ALLOW - UID(*) SERVICE(ADD) ALLOW
DBR	* DLV@BANR BANNERS \$KEY(RMO) TYPE(DBR) - UID(*) SERVICE(ADD) ALLOW
DBS	* DLV@DBAS DATABASE \$KEY(RMO) TYPE(DBS) - UID(*) SERVICE(ADD) ALLOW

Member	Contents
DDI	* DLV@DIST DISTIDS \$KEY(RMO) TYPE(DDI) UID(*) SERVICE(READ) ALLOW - UID(*) SERVICE(ADD) ALLOW
DJB	* DLV@JOB JOBS \$KEY(RMO) TYPE(DJB) UID(*) SERVICE(READ) ALLOW - UID(*) SERVICE(ADD) ALLOW
DPN	* DLV@PANL PANELS \$KEY(RMO) TYPE(DPN) - UID(*) SERVICE(ADD) ALLOW
DRP	* DLV@REPT REPORTS \$KEY(RMO) TYPE(DRP) UID(*) SERVICE(READ) ALLOW - UID(*) SERVICE(ADD) ALLOW

Bypassing Password Verification

For the following online interfaces, you can bypass password specification and allow a user to log on to a CA Deliver database once that user is logged on to the online interface:

- CICS pseudo-conversational
- IMS
- ISPF/cross-memory
- TSO/cross-memory
- CA Roscoe/cross-memory

To implement this functionality, follow these steps:

1. Specify LGNSEC=YES for the cross-memory region startup parameter.
2. Specify the appropriate SAFPROT rules for CA ACF2.

Note: For information about the LGNSEC parameter, see the *Installation Guide*. For CA ACF2, see the following sample SAFPROT rules (additional modification can be required due to site requirements):

For the ISPF interface:

```
SAFPROT.RM01CLASSES(DATASET)CNTLPTS(RM0SPF)SUBSYS(SVC019)
SAFPROT.RM02CLASSES(DATASET)CNTLPTS(RM0SPF)SUBSYS(RM0SPF)
```

For the VTAM/XMEM interface:

```
SAFPROT.RM03CLASSES(DATASET)CNTLPTS(EB0XMDRV)SUBSYS(SVC019)
SAFPROT.RM04CLASSES(DATASET)CNTLPTS(EB0VTDRV)SUBSYS(SVC019)
SAFPROT.RM05CLASSES(DATASET)CNTLPTS(EB0XMDRV)SUBSYS(EB0XMDRV)
SAFPROT.RM06CLASSES(DATASET)CNTLPTS(EB0VTDRV)SUBSYS(EB0VTDRV)
```

Implementing External Security for RACF

Note: For more information about the RACF Class Descriptor Table and the Routing Table, see SC28-1913 in IBM's *OS/390 Security Server (RACF) System Programmer's Guide*. For more information about the commands used in this section, see SC28-1919 in the *OS/390 Security Server (RACF) Command Language Reference*. (At OS/390 V2R10.0, OS/390 Security Server (RACF) has been renamed to OS/390 SecureWay Security Server RACF.) The sample jobs can be found in CVDEJCL member RMORACF.

To use RACF to manage CA Deliver external security, follow these steps:

1. Create or add code to the RACF Class Descriptor Table.

For example, the following job creates a Class Descriptor Table that contains the CA Deliver nine class names. The table must be assembled and linked as ICHRRCDE. If you have already created one of these tables, include it in the link step. Otherwise, remove the INCLUDE SYSLMOD(ICHRRCDE) statement from the link step.

```
//EXAMPLE JOB ACCOUNT,PROGRAMMER
//CDT EXEC HLASMCL
//C.SYSLIB DD DSN=SYS1.MODGEN,DISP=SHR
//C.SYSIN DD *

DLV@ACT ICHERCDE CLASS=DLV@ACT,ID=128,MAXLNTH=37,FIRST=ALPHA,      +
        OTHER=ANY,POSIT=25,OPER=NO
DLV@BACT ICHERCDE CLASS=DLV@BACT,ID=128,MAXLNTH=37,FIRST=ALPHA,    +
        OTHER=ANY,POSIT=25,OPER=NO
DLV@BANR ICHERCDE CLASS=DLV@BANR,ID=128,MAXLNTH=13,FIRST=ALPHA,    +
        OTHER=ANY,POSIT=25,OPER=NO
DLV@BNDL ICHERCDE CLASS=DLV@BNDL,ID=128,MAXLNTH=37,FIRST=ALPHA,    +
        OTHER=ANY,POSIT=25,OPER=NO
```

```

DLV@DBAS ICHERCDE CLASS=DLV@DBAS, ID=128, MAXLNTH=22, FIRST=ALPHA,      +
        OTHER=ANY, POSIT=25, OPER=NO
DLV@DIST ICHERCDE CLASS=DLV@DIST, ID=128, MAXLNTH=37, FIRST=ALPHA,      +
        OTHER=ANY, POSIT=25, OPER=NO
DLV@JOB  ICHERCDE CLASS=DLV@JOB, ID=128, MAXLNTH=13, FIRST=ALPHA,       +
        OTHER=ANY, POSIT=25, OPER=NO
DLV@PANL ICHERCDE CLASS=DLV@PANL, ID=128, MAXLNTH=13, FIRST=ALPHA,      +
        OTHER=ANY, POSIT=25, OPER=NO
DLV@REPT ICHERCDE CLASS=DLV@REPT, ID=128, MAXLNTH=37, FIRST=ALPHA,      +
        OTHER=ANY, POSIT=25, OPER=NO
        ICHERCDE
/*

//L.SYSLMOD DD DSN=SYS1.LINKLIB,
//          DISP=SHR
//L.SYSIN   DD *
        INCLUDE SYSLMOD(ICHRRUDE) NEEDED IF ADDING TO AN EXISTING TABLE
        ORDER  DLV@ACT
        ORDER  DLV@BACT
        ORDER  DLV@BANR
        ORDER  DLV@BNDL
        ORDER  DLV@DBAS
        ORDER  DLV@DIST
        ORDER  DLV@JOB
        ORDER  DLV@PANL
        ORDER  DLV@REPT
        ORDER  ICHRRUDE
        NAME   ICHRRUDE(R)
/*

```

2. Add the CA Deliver class names to the RACF Router Table, for example:

```

//EXAMPLE  JOB ACCOUNT, PROGRAMMER
//RT  EXEC HLASMCL
//C.SYSLIB DD DSN=SYS1.MODGEN, DISP=SHR
//C.SYSIN DD *
ICHRFR01 CSECT
DLV@ACT  ICHRFRTB CLASS=DLV@ACT, ACTION=RACF
DLV@BACT ICHRFRTB CLASS=DLV@BACT, ACTION=RACF
DLV@BANR ICHRFRTB CLASS=DLV@BANR, ACTION=RACF
DLV@BNDL ICHRFRTB CLASS=DLV@BNDL, ACTION=RACF
DLV@DBAS ICHRFRTB CLASS=DLV@DBAS, ACTION=RACF
DLV@DIST ICHRFRTB CLASS=DLV@DIST, ACTION=RACF
DLV@JOB  ICHRFRTB CLASS=DLV@JOB, ACTION=RACF
DLV@PANL ICHRFRTB CLASS=DLV@PANL, ACTION=RACF
DLV@REPT ICHRFRTB CLASS=DLV@REPT, ACTION=RACF
ENDTAB  ICHRFRTB TYPE=END
        END  ICHRFRT01
/*

```

```
//L.SYSLMOD DD DSN=SYS1.LINKLIB,  
//          DISP=SHR  
//L.SYSIN   DD *  
          NAME   ICHRF01(R)  
/*
```

3. Activate the new classes, for example:

```
//EXAMPLE   JOB ACCOUNT,PROGRAMMER  
//CLSA EXEC PGM=IKJEFT01  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *  
SETR CLASSACT(DLV@ACT)  
SETR CLASSACT(DLV@BACT)  
SETR CLASSACT(DLV@BANR)  
SETR CLASSACT(DLV@BNDL)  
SETR CLASSACT(DLV@DBAS)  
SETR CLASSACT(DLV@DIST)  
SETR CLASSACT(DLV@JOB)  
SETR CLASSACT(DLV@PANL)  
SETR CLASSACT(DLV@REPT)  
/*
```

4. Define a group to own the resources, for example:

```
//EXAMPLE   JOB ACCOUNT,PROGRAMMER  
//AG EXEC PGM=IKJEFT01  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *  
AG (DLVRADMN) OWNER(SYS1) SUPGROUP(SYS1)  
/*
```

5. To give READ access to all of the functions and ALTER access to all of the resources, run the following job steps:

```
//EXAMPLE   JOB ACCOUNT,PROGRAMMER  
//RDEF EXEC PGM=IKJEFT01  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *  
RDEF DLV@ACT (RMO.) OWNER(DLVRADMN) UACC(READ)  
RDEF DLV@BACT (RMO.) OWNER(DLVRADMN) UACC(READ)  
RDEF DLV@BNDL (RMO.) OWNER(DLVRADMN) UACC(READ)  
RDEF DLV@DIST (RMO.) OWNER(DLVRADMN) UACC(READ)  
RDEF DLV@JOB (RMO.) OWNER(DLVRADMN) UACC(READ)  
RDEF DLV@REPT (RMO.) OWNER(DLVRADMN) UACC(READ)  
/*  
  
//EXAMPLE   JOB ACCOUNT,PROGRAMMER  
//RDEF EXEC PGM=IKJEFT01  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *
```

```

RDEF DLV@ACT (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@BACT (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@BANR (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@BNDL (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@DBAS (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@DIST (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@JOB (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@PANL (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
RDEF DLV@REPT (RMO.*) OWNER(DLVRADMN) UACC(ALTER)
/*

```

6. Connect a user to the group and alter the user definition (so its default group is the one you now created), for example:

```

//EXAMPLE JOB ACCOUNT,PROGRAMMER
//CONN EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
CO (userid) GROUP(DLVRADMN)
/*

//EXAMPLE JOB ACCOUNT,PROGRAMMER
//ALU EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ALU (userid) DFLTGRP(DLVRADMN)
/*

```

Bypassing Password Verification

For the following online interfaces, you can bypass password specification and allow a user to log in to a CA Deliver database after that user is logged on to the online interface:

- CICS pseudo-conversational
- IMS
- ISPF/cross-memory
- TSO/cross-memory
- CA Roscoe/cross-memory

To implement this functionality, follow these steps:

1. Specify LGNSEC=YES for the cross-memory region startup parameter.
2. Define the RMOXMS region to RACF, using the ICHRIN03 MVS table.

After setting up the RMOXMS region definition and assembling it into the linklist libraries, the following RACF commands must be executed on TSO/RACF:

```
AG (PROCGRP) OWNER (SYS1) SUPGROUP (SYS1)
AU (RMO60) PASSWORD (PASS) OWNER (PROCGRP) DFLTGRP (PROCGRP)
```

You must IPL to implement these changes.

Note: For more information about the LGNSEC parameter, see the *Installation Guide*.

Appendix A: CA Deliver Health Checks

This Appendix describes health checks for CA Deliver and the informational (I) or exception (E) messages that are generated. The product owner for all CA Deliver health checks is CA_DLVR.

This section contains the following topics:

[The Health Checks](#) (see page 343)

[Messages](#) (see page 345)

The Health Checks

DLVR_OPT_HDETAIL

Description

This health check allows you to ensure that the HDETAIL option is only activated when necessary. With the CA Deliver HDETAIL option, you are doing station tracking of all reports processed by this CA Deliver database. This feature has a high overhead and can be used only for a small CA Deliver database.

When this option is turned on, database space usage rapidly grows and jobs running in pre-spool mode experience delays as the history records are created for each recipient of the report.

Best Practice

Determine if HDETAIL was intentionally turned on. Also verify that the two CA Deliver JES exits have been installed. If these conditions have not been met, set HDETAIL=NO in the RMOPARMS data set and cycle the RMOSTC task.

Parameters Accepted

No

Debug Support

No

Verbose Support

Yes

Severity

Low

Interval

Once

Reference

See the following:

- CA Deliver Installation Guide for information about installing the CA Deliver JES Printer exits.
- CA Deliver Reference Guide for information about how to set the HDETAIL parameter.
- CA Deliver Administrator Guide for information about online and batch tracking commands.

Messages

The following messages are generated:

- RMOH002E
- RMOH002I

DLVR_PRFM_PQE

Description

This health check reports on queued requests. A large number of queued requests indicate that something is degrading CA Deliver performance. A common cause is delayed access to the CA Deliver database or checkpoint data sets. Workload spikes are normal, but they must be short lived. If this condition occurs frequently, verify that a long-term reserve is not occurring on the CA Deliver checkpoint or database volumes.

Best Practice

Monitor CA Deliver application work queues for large queue sizes, which indicate potential delays. CA Deliver processes application work requests through PRBTASKS. Because these requests are from multiple address spaces and can occur concurrently, work request queues are maintained in CSA. Typically, CA Deliver must be able to keep the queue size under 100.

You can also increase the CA Deliver PRBTASKS parameter up to a maximum of 16.

Note: Changing this parameter requires the CA Deliver started task to be cycled.

Parameters Accepted

Yes, WARN=256

Debug Support

No

Verbose Support

Yes

Severity

Medium

Interval

20 minutes

Exception

4 minutes

Reference

See the CA Deliver Reference Guide for information about the PRBTASKS parameter.

Messages

The following messages are generated:

- RMOH001E
- RMOH001I

Messages

Health checks generate the following messages.

RMOH001E

CA Deliver work request backlog is currently at xxxx. This exceeds the WARNING threshold of xxxx.

Reason:

The CA Deliver processes application work the requests using PRBTASKS. Because these requests are from multiple address spaces and can occur concurrently, work request queues are maintained in CSA. Typically, CA Deliver must be able to keep the queue size under 100.

A large number of queued requests indicates that something is degrading CA Deliver performance. A common cause is delayed access to the CA Deliver database or checkpoint data sets. Verify that a long term reserve is not occurring on the CA Deliver checkpoint or database volumes.

Action:

Notify the systems programmer of this condition and verify that a long-term reserve is not occurring on the CA Deliver checkpoint or database volumes. You can also increase the CA Deliver PRBTASKS parameter up to a maximum of 16.

Changing this parameter requires the CA Deliver started task to be cycled. CA Deliver continues processing.

RM0H001I

CA Deliver work queues are operating normally. Number of requests queued: nnnnnn, Highest no. reqs. Queued: nnnnnn

Reason:

Informational message stating that the requests from application address spaces are not experiencing any delays.

Action:

None.

RM0H002E

CA Deliver is running with HDETAIL=YES.

Reason:

The CA Deliver HDETAIL option allows clients to implement station tracking for all reports processed by this CA Deliver database. This feature has a high overhead and can be used only for a small CA Deliver database. When this option is turned on, database space usage rapidly grows and jobs running in pre-spool mode experience delays as the history records are created for each recipient of the report.

Action:

Contact the administrator responsible for this CA Deliver database. Determine if HDETAIL was intentionally turned on. Also verify that the two Deliver JES exits have been installed. If all of these conditions have not been met, set HDETAIL=NO in the RMOPARMS data set and cycle the RMOSTC task CA Deliver continues processing.

RM0H002I

CA Deliver is not collecting detail history.

Reason:

Informational message. Collecting detail history incurs additional overhead and database space; most clients do not require this option. Turning off history detail does not stop basic history from be collected on the CA Deliver database. Basic history includes Date, Time, Job ID, pages and lines for each report and bundle.

Action:

Contact the administrator responsible if you want to turn on this option.

Index

\$

\$\$FIRST • 244
\$BANNER • 244

/

/*FORMAT JES3 control statement • 33
//OUTPUT control statement • 30
/ATTR control statement • 289
/BEGSEP control statement • 288
/END control statement • 289
/ENDSEP control statement • 289
/REPORT control statement • 307

A

access • 20, 314
 levels of security • 314
 method control block • 20
ACCOUNT operand • 316
activating reports • 250
active • 85, 167, 264
 bundle queue, scanning an • 85
 bundle status record • 167
 definitions, updating • 264
 report status record • 167
Active Bundle List report • 261
Active Bundle Status Record, database fields for • 189
Active Report List report • 261
Active Report panel • 264
Active Report Status Record, database fields for • 188
ADDDDS • 106, 108
 control statement • 106
 parameters, interdependency of • 108
AFP commands • 243
American Standards Association (ASA) • 303
appending printer setup records to a report • 303
ASA (American Standards Association) • 303
assigning an identifier • 27
attribute characters, definition of • 287
attributes • 27, 272
 database data set • 272
 network input • 27
authorization • 313, 316, 318

group criteria • 318
macros • 316
user exit (RMOATHUX) • 313
AUTHTID initialization parameter • 313
automatic • 229
 database construction from JCL • 229

B

Banner Member Control Record (BCR) • 273
banner page • 30, 244, 281, 286
 for monitored data output • 30
 library, definition of • 281
 members, types of • 286
 problems, correcting • 244
Banner Page Member List report • 261
banners • 244
 TSTBNDL • 244
 TSTDIST • 244
 TSTRPT • 244
basic report processing • 28
batch • 99, 239, 258
 bundle posting • 99
 reporting • 258
 utility • 239
Batch Detail History Reporting (RMOHTP) • 225
BCR (Banner Member Control Record) • 273
BDR (Bundle Descriptor Record) • 273
BHR (Bundle Historical Record) • 273
BLKSIZE parameter • 108
BLOAD control statement • 109
BNDLBNRn initialization parameter • 286
BNDLDEF control statement • 131
BNDLSCAN initialization parameter • 85
BREAK control statement • 196
buffered data • 85
buffers, refreshing • 85
bundle • 27, 37, 84, 167, 244, 275
 data • 275
 definition attributes online panel • 244
 definitions • 275
 descriptor record • 167
 detail history record • 167
 history record • 167
 holding copy attributes • 27
 output task (BOT) • 37, 84

- Bundle Definition Table • 85
- Bundle Descriptor Record (BDR) • 273
- Bundle Detail History Record • 187
- Bundle Distribution Cross-Reference Record (XBR) • 273
- Bundle Historical Record (BHR) • 273
- Bundle History Record • 186
- Bundle Listing by Bundle Identifier report • 261
- bundles, freeing from a system • 84

C

- CA 11 • 240, 255
 - setting up • 240
 - using the product with • 255
- CA ACF2 • 333
- CA Balancing • 84, 89, 242
 - database • 89
 - rerun interface routine • 242
 - starting/stopping • 84
- CA JCLCheck • 234
- CA Librarian library • 237
- CA L-Serv • 242
- CA Output Management Document Viewer, sending reports/bundles to • 312
- CA Panvalet library • 237
- CA Top Secret • 331
- CA View, ACIF interface • 240
- CAI.CAILIB • 229
- CAI.CAIMAC • 273
- calculating cylinders for Checkpoint • 118
- calculating cylinders for database • 108
- CANCEL command • 89
- carriage control • 30, 290
 - characters • 290
- catalogued procedure libraries • 211
- characters • 287, 290
 - attribute • 287
 - carriage control • 290
- checkpoint • 87, 103, 117, 118, 274, 275
 - bundle entry (BCE) • 103
 - calculating cylinders • 118
 - contents report • 103
 - creation of • 117
 - data set • 274, 275
 - attributes • 275
 - structure of • 274
 - report • 103
 - bundle entry (CRB) • 103
 - job entry (CRJ) • 103
 - unlocking the • 87
- Checkpoint Contents Report • 103
- Checkpoint Map utility • 100
- Checkpoint, types of records in • 103
- CLOSE command • 90
- close processing • 20
- codes, error • 281
- COLLECT control statement • 165
- commands • 85, 243
 - AFP • 243
 - RESET • 85
- condition, definition of • 195
- constructing a database from existing data • 127
- continuation lines • 316, 322
- CONTINUE control statement • 196
- control • 30, 303
 - break identification, definition of • 30
 - break report processing • 30
 - records, X'5A' • 303
- CONTROL control statement • 197
- control statements • 30, 33, 97, 98, 106, 109, 111, 112, 113, 114, 115, 117, 119, 120, 121, 123, 129, 131, 135, 168, 196, 197, 199, 201, 203, 204, 205, 208, 210, 212, 213, 214, 215, 226, 227, 231, 232, 234, 238, 239, 240, 241, 242, 244, 246, 248, 249, 252, 253, 256, 257, 259, 260, 263, 265, 267, 268, 269, 277, 288, 289, 307
 - /*FORMAT • 33
 - //OUTPUT • 30
 - /ATTR • 289
 - /BEGSEP • 288
 - /END • 289
 - /ENDSEP • 289
 - /REPORT • 307
- ADD DS • 106
- ASCII • 246
- BLOAD • 109
- BNDLDEF • 131
- BREAK • 196
- CONTINUE • 196
- CONTROL • 197
- CONVERT • 111
- COPY • 112, 265
- DATA • 226
- DATE • 227
- DBASE • 129
- DEFINE • 199
- definition of • 97

DELBAN • 113
DELETE • 114, 267
DELPAN • 114
DELPRSET • 114
DISTDEF • 135
DO • 201
ELSE • 203
END • 203
EXCLUDE • 232, 238, 242, 249
 explanation of • 277
 for model banner page members • 288
FROM-NODE • 256
FROM-STEP • 241
HDELETE • 115
hexadecimal • 246
IF • 204
job • 129, 168, 226, 231, 240, 248, 252, 253, 256,
 259, 263, 265, 277
 name translation table • 277
 RMODBB • 129
 RMOGRW • 168
 RMOHTP • 226
 RMOJCL • 231
 RMOPRE • 240
 RMOPSF • 248
 RMORAP • 252
 RMORMS • 256
 RMORPT • 259
 RMORXB • 263
 RMOUTIL • 265
JOBNAME • 240
JOBNUM • 241
LOAD • 115
LSERV-INBSSN • 242, 257
MAKECKPT • 117
MDLFDEF • 244, 249
MDLPDEF • 244, 249
NAME • 119, 227, 232, 238, 240, 253, 256, 260,
 268
NEXT • 205
OLOAD • 119
OMIT • 232, 238
ON • 205
PLOAD • 120
PRINT • 208
RELEASE • 210
REMOVE • 249
RENAME • 120, 269
REPORT • 253, 260

rerun • 240
SELECT • 210, 234, 239, 249
SET • 212
SORT • 213
STATUS • 121
STOP • 214
 syntax rules for • 98
THEN • 215
TIME • 227
TITLE • 215
TO-STEP • 241
TYPE • 253
UNLOAD • 123
CONVERT control statement • 111
converted block letters • 288
COPY • 112, 265
 control statement • 112, 265
 copy modification module • 300
 cross-memory • 88, 89, 90, 91
 task • 88, 89, 90, 91
 canceling users from • 89
 closing • 90
 listing statistics for • 91
 normal termination • 88
 opening • 91
 shutting down • 88
 starting • 88
 suspending users from • 90
 cross-reference records, rebuilding • 263
CYLINDER parameter • 108
cylinders, calculating • 108

D

data • 85, 228, 246, 258
 definition name field • 258
 map • 246
 refreshing buffered • 85
 station • 228
DATA control statement • 226
database • 21, 127, 234, 266, 271, 272, 273
 constructing from existing data • 127
 construction • 234
 data set attributes • 272
 definition of • 271
 records • 266, 273
 copy restrictions and limitations • 266
DATACLAS parameter • 108
DATE control statement • 227

DBASE control statement • 129
DD • 228, 258
 NAME field • 258
 statements, STATION • 228
DDR (Distribution Descriptor Record) • 273
deactivating reports • 250
DEFINE control statement • 199
defining a database • 27
definition data • 25
definitions • 243
 form • 243
 page • 243
 page and form • 243
DELBAN control statement • 113
DELETE control statement • 114, 267
DELPAN control statement • 114
DELPRSET control statement • 114
diagram of operation • 18
DISTDEF control statement • 135
distribution • 85, 167, 260, 261
 data entries • 85
 data record • 167
 identifiers • 260, 261
 report identifier cross-reference • 261
 report of distribution detail • 260
Distribution Data Record, database fields for • 182
Distribution Descriptor Record (DDR) • 273
Distribution List Cross-Reference Record (XGR) • 273
DJDE (Dynamic Job Descriptor Entries) • 304, 305
 coding statements • 304
 generating • 304
 identification • 305
 offset and prefix • 305
DO control statement • 201
dynamic • 19
 interface • 19
Dynamic Job Descriptor Entries (DJDE) • 304, 305
 coding statements • 304
 generating • 304
 identification • 305
 offset and prefix • 305

E

ELSE control statement • 203
END control statement • 203
enhanced database construction from JCL • 234
entries • 304
 PDE1 • 304

 PDE4 • 304
EXCLUDE control statement • 232, 238, 242, 249
EXEC JCL statement • 240
execution job name • 278
exits, userSee user exits • 230
expression, definition of • 191
external security • 326

F

F operator command • 83
F operator commandFopercommands • 84
FCB image identification • 300
features, new, in r11 • 16
field names • 170
fields • 190, 258
 DD NAME • 258
 JOB STEP • 258
 JOBID • 258
 PROC STEP • 258
 REPORT ID • 258
 reserved • 190
 STATUS • 258
form • 243, 246
 and page definition modification • 246
 definition • 243
FORMDEF • 244, 246
 contents of • 246
 members • 244
forms overlay • 300
freeing reports and bundles from a system • 84
FROM-NODE control statement • 256
FROM-STEP control statement • 241
FUNCTION keyword • 156

G

general report writer (RMOGRW) • 167, 170
 description of • 167
 using field names • 170
generic identifiers • 321
GREXX variables • 308

H

HDELETE control statement • 115
health checks • 343
historical data • 30, 275

I

IBM • 303

- 3800 printer • 303
- IEBGENER • 101
- IF control statement • 204
- image identification • 300
- inactivating reports • 250
- Index File Mapping Utility • 229
- initialization parameters • 21, 35, 36, 37, 85, 86, 286, 313
 - ARCH • 86
 - AUTHTID • 313
 - BANNER • 86
 - BEGINDAY • 86
 - BNDLBNRn • 86, 286
 - BNDLCONF • 86
 - BNDLHDTL • 86
 - BNDLINT • 86
 - BNDLMOUT • 37
 - BNDLSCAN • 37, 85, 86
 - BNDLWAIT • 86
 - BOT • 35, 37
 - DAYS • 86
 - FREEALL • 86
 - GSS • 86
 - HDETAIL • 86
 - JOBCLSL • 36, 86
 - LOGO • 86
 - MAXHIST • 86
 - NETCLSL • 21
 - NETDEST • 21
 - NETFORM • 21
 - setting while the product is running • 86
 - SMF • 86
 - START • 86
 - STKCHNn • 86
 - STKMODE • 86
 - STNAMEn • 86
 - TEXT • 86
 - TIME • 86
- interfacing with CA View • 17
- interleaved report processing • 29
- internal • 244, 313
 - labels (\$BANNER and \$\$FIRST) • 244
 - security • 313

J

- JCL • 229, 234
 - automatic database construction from • 229
 - enhanced database construction from • 234

- JCLCheck common component • 234
- JCR (Job Descriptor Record) • 273
- JES • 21, 25, 241
 - job number • 241
 - spool • 21
- JES3,/*FORMAT control statement • 33
- JFCB (Job File Control Block) • 25
- job • 19, 100, 105, 129, 167, 168, 226, 231, 240, 248, 252, 256, 258, 259, 265, 277
 - control statements • 100, 105, 129, 168, 226, 231, 240, 248, 252, 256, 259, 265
 - RMOBPR • 100
 - RMODBASE • 105
 - RMODBB • 129
 - RMOGRW • 168
 - RMOHTP • 226
 - RMOJCL • 231
 - RMOPRE • 240
 - RMOPSF • 248
 - RMORAP • 252
 - RMORMS • 256
 - RMORPT • 259
 - RMOUTIL • 265
 - descriptor record • 167
 - identifier field • 258
 - name translation table • 277
 - number field • 258
 - step field • 258
 - step name • 19
- JOB control statement • 253
- Job Descriptor Record (JCR) • 273
- Job File Control Block (JFCB) • 25
- JOBCLSL initialization parameter • 19, 36
- JOBNAME control statement • 240
- Jobname Translation Table • 85
- JOBNUM control statement • 241
- Julian dates, interpreting • 170

K

- keywords • 156, 231
 - CHECK • 231
 - FUNCTION • 156
 - TEST • 231

L

- labels, internal • 244
- LGNSec parameter • 337, 341
- libraries, model banner page • 281

LIST • 91
 command • 91
LOAD control statement • 115
loading printer setup library members • 303
LSERV-INBSSN control statement • 242, 257

M

macros • 273, 316
 authorization • 316
 coding order of • 316
 mapping to records • 273
MAKECKPT control statement • 117
mapping macros to records • 273
Master Control Record (MCR) • 273
MCR (Master Control Record) • 273
MDLFDEF control statement • 244, 249
MDLPDEF control statement • 244, 249
members • 244, 286
 FORMDEF • 244
 PAGEDEF • 244
 types of model banner page • 286
migration support • 264
model banner page • 281, 286
 library, definition of • 281
 member, definition of • 281
 members, types of • 286
monitored data output • 30
multi-node network • 21
multi-sysplexes • 24

N

NAME • 119, 227, 232, 238, 240, 253, 256, 260, 268
 control statement • 119, 227, 232, 238, 240, 253, 256, 260, 268
NAME control statement • 165
NETCLSL initialization parameter • 21, 23
NETDEST initialization parameter • 21, 23
NETFORM initialization parameter • 21, 23
network input • 21, 27
 attributes • 27
NEXT control statement • 205
NJE unattended download • 311
non-shared spool • 22
non-SMS data sets • 108, 118

O

obtaining print attributes • 31
OCR (Online Member Control Record) • 273

OLOAD control statement • 119
OMIT • 232, 233, 238
 control statement • 232, 238
 function name • 233
ON control statement • 205
Online Member Control Record (OCR) • 273
Online Panel Member List report • 261
OPEN command • 91
open processing • 19
OPER operand • 316
operands • 123, 316
 ACCOUNT • 316
 BLKSIZE • 123
 OPER • 316
 USER • 316
operator commands • 82, 83, 84, 85
 F • 83
 FopercommandsF • 84
 P • 83
 S • 82, 83
 SET • 85
OUTPUT initialization parameter • 30
OUTPUT JCL statement • 25

P

P operator command • 83
page • 243
 and form definitions • 243
 definition • 243
Page Descriptor Entry (PDE) • 304
PAGEDEF • 244, 246
 contents of • 246
 members • 244
panels • 286
 Bundle Definition Attributes • 286
 Report Definition Attributes • 286
parameters • 85, 86, 240
 displaying currently active • 85
 initialization • 86
 setting while the product is running • 86
 PARM • 240
PARM parameter • 100, 240
password verification • 337, 341
PDDDB (Peripheral Data Definition Blocks) • 25
PDE (Page Descriptor Entry) • 304
PDE1 entries • 304
PDE4 entries • 304
PDS members • 336

Peripheral Data Definition Blocks (PDDDB) • 25
PLOAD control statement • 120
posting the bundle output task • 85
post-spool • 20, 21
 operations • 21
 processing • 20
pre-spool operation • 23
print • 30, 31
 attributes • 30, 31
PRINT control statement • 208
Print Services Facility (PSF) • 303
printer • 303
 setup • 303
 library members, loading • 303
 library, definition of • 303
 member, definition of • 303
Printer Setup Member List report • 261
printers • 303, 304
 IBM 3800 • 303
 Xerox 9700 • 303, 304
PROC name • 234
procedure • 19, 258
 step field • 258
 step name • 19
Process SYSOUT interface (PSO) • 21
production • 211
 JCL library • 211
 jobs • 211
PRSET entries • 85
PRSET field • 303, 305
 optional use of • 305
PSF (Print Services Facility) • 303
PSO (Process SYSOUT interface) • 21

Q

QUIESCE command • 88

R

RACF Class Descriptor Table • 338
RCE checkpoint report entry • 103
RCS (Report Control Statements) • 273
RDR (Report Descriptor Record) • 273
rebuilding cross-reference records • 263
records, type 6 SMF • 299
REFRESH parameter • 82
refreshing buffers • 85
RELEASE control statement • 210
REMOVE control statement • 249

RENAME control statement • 120, 269
report • 27, 29, 167, 258, 260, 275, 277
 attribute definitions • 277
 definitions • 275
 descriptor record • 167
 detail history record • 167
 historical record • 167
 identification text, definition of • 29
 identifier field • 258
 listing • 260
 by job • 260
 by report identifier • 260
 processing • 27
REPORT control statement • 253, 260
Report Control Statements (RCS) • 273
Report Data Set Collector, RMODSC • 163
Report Definition Attributes panel • 244, 286
Report Descriptor Record (RDR) • 273
Report Detail History Record, database fields for • 181
Report Distribution Cross-Reference Record (XDR) • 273
Report Historical Record (RHR) • 273
reporting, batch • 258
reports • 84, 250, 257, 261
 activating • 250
 Active Bundle List • 261
 Active Report List • 261
 Banner Page Member List • 261
 Bundle Listing by Bundle Identifier • 261
 freeing from a system • 84
 inactivating • 250
 Online Panel Member List • 261
 Printer Setup Member List • 261
 Rerun Processing Status • 257
rerun control statements • 240
reserved fields • 190
Resource Descriptor Table • 331
REXX EXEC, example of • 309
RHR (Report Historical Record) • 273
RMOACF2 • 333
RMOAEND macro • 316
RMOAGRP macro • 316
RMOATH macro • 316
RMOATHTB • 313
RMOATHUX • 313
RMOBCR • 273
RMOBDR • 273
RMOBHE • 273

RMOBHR • 273
RMOBPR batch utility • 99
RMOCPPMAP • 100, 102
RMOCPTIN DD statement • 101
RMODBASE utility • 104, 271
RMODBB • 98, 127, 129
RMODDR • 273
RMODSC Report Data Set Collector • 163
RMOGRW (general report writer) • 167, 170
 description of • 167
 using field names • 170
RMOHTP • 225, 226
 JCL for • 226
RMOIFMAP • 229
RMOJCL • 98, 229, 231
RMOJCLUX • 230, 234
RMOJCS • 234
RMOJDR • 273
RMOJTAB • 277
RMOMCR • 273
RMOOCR • 273
RMOOLIB • 119
RMOPLIB • 120
RMOPRE • 239, 240
RMOPSF • 243, 244, 246, 248
RMORACF • 338
RMORAP • 250, 252
RMORCS • 273
RMORDR • 273
RMORHE • 273
RMORHR • 273
RMORMS • 255, 256
RMORPT • 258, 259
RMORXB • 263
RMOSMFUX • 299
RMOSTC • 82
RMOSTC RESET command • 85
RMOTSS • 331
RMOUCR • 273
RMOUNLD • 123
RMOUTIL • 264, 265
RMOXBR • 273
RMOXDR • 273
RMOXGR • 273
RMOXMS (cross-memory task) • 87, 92
Routing Table • 338

S

S operator command • 82, 83
SAF calls • 333
scanning an active bundle queue • 85
Schedular Work Block (SWB) • 30
security • 313, 314, 321, 326
 access levels, types of • 314
 external • 326
 internal • 313
 table • 313, 321
 RMOATHTB • 313
SELECT control statement • 210, 234, 239, 249
SEPARATOR field • 286
service routines • 82
SET • 85, 212
 control statement • 212
 operator command • 85
shared spool environment • 23
SI/OT (Step Input/Output Table) • 25
SMF (System Management Facilities) • 299
SMS data sets • 108, 118
SORT control statement • 213
space analysis, Checkpoint • 102
stacked report processing • 28
started task • 19, 20, 25, 27
 procedure • 27
starting • 82, 84, 88
 CA Balancing • 84
 CA Deliver • 82
 cross-memory task • 88
 the bundle output task • 84
station data • 228
status • 257, 258
 field • 258
 report, rerun processing • 257
STATUS control statement • 121
STC name • 82
Step Input/Output Table (SI/OT) • 25
STKMODE initialization parameter • 30
STOP control statement • 214
stopping • 83, 84, 88
 CA Balancing • 84
 CA Deliver • 83
 cross-memory task • 88
 the bundle output task • 84
STORCLAS parameter • 108
sub-panel, Control Break Identification • 286
substituted text • 288

- subsystem • 300
 - identification • 300
- SUSPEND command • 90
- SWB (Schedular Work Block) • 30
- symbolic variable • 287, 291, 292
 - defining the start and end of a • 287
 - definition of a • 291
 - example of a • 291
 - types of • 292
- syntax • 98
 - rules for using control statements • 98
- SYSCLSL initialization parameter • 19
- SYSID initialization parameter • 27
- SYSOUT • 20, 21, 26, 233
 - attributes • 21
 - class • 233
 - data set • 20, 26
- sysplexes • 24
- SYSPRINT DD statement • 233
- System Management Facilities (SMF) • 299
- System Requirements
 - routines, refreshing • 82

T

- table • 25, 85, 277, 331, 338
 - Bundle Definition • 85
 - Job name Translation • 277
 - Jobname Translation • 85
 - RACF Class Descriptor • 338
 - Resource Descriptor • 331
 - Routing • 338
 - Step Input/Output (SI/OT) • 25
- text • 288
 - substituting • 288
- THEN control statement • 215
- TIME • 227
 - control statement • 227
- TITLE control statement • 215
- TO-STEP control statement • 241
- tracking data • 275
- translated job name • 278
- type 6 SMF records • 299
- TYPE control statement • 253

U

- UCR (User Attribute Control Record) • 273
- UCS image identification • 300
- unattended downloading • 311

- UNIT parameter • 108
- UNLOAD control statement • 123
- unlocking the checkpoint • 87
- definitions • 128
- user • 316
 - IDs, security systems and • 316
- User Attribute Control Record (UCR) • 273
- user exits • 230, 299, 313
 - RMOATHUX • 313
 - RMOJCLUX • 230
 - RMOSMFUX • 299
- USER operand • 316
- using CA 11 with the product • 255
- utilities • 98, 99, 100, 104, 127, 163, 167, 225, 229, 234, 239, 246, 250, 255, 258, 263, 264, 271
 - RMOBPR • 99
 - RMOCPMAP • 100
 - RMODBASE • 104, 271
 - RMODBB • 98, 127
 - RMODSC • 163
 - RMOGRW • 167
 - RMOHTP • 225
 - RMOIFMAP • 229
 - RMOJCL • 98, 229
 - RMOJCS • 234
 - RMOPRE • 239
 - RMOPSF • 246
 - RMORAP • 250
 - RMORMS • 255
 - RMORPT • 258
 - RMORXB • 263
 - RMOUTIL • 264

V

- VOLSER parameter • 108
- VTAM • 93
 - multiple cross-memory regions • 93

W

- withdrawing CA Deliver from an operating system • 83

X

- X'5A' (Hex Five Able) • 243, 303
 - control records • 303
- XBR (Bundle Distribution Cross-Reference Record) • 273

XDR (Report Distribution Cross-Reference Record) •

273

Xerox 9700 printer • 303, 304

XGR (Distribution List Cross-Reference Record) • 273